REVISTA DE TECNOLOGÍA E INFORMÁTICA HISTÓRICA

Volumen 5 Número 1 2015

ARTÍCULOS

ISSN 2250-5431

Revista de Tecnología e Informática Histórica
Volumen 5, Número 1, 2015

Diego G. Macchi
Editor responsable



Revista de Tecnología e Informática Histórica es una publicación periódica propiedad de la Fundación Museo ICATEC. Museo de Informática, Computadoras y Accesorios Tecnológicos.
Tucumán 810 1er. Subsuelo, CP C1049AAR, Capital Federal.

Registro intelectual en trámite

LA COMPUTADORA DE LAS MISIONES APOLO
PARTE III: SOFTWARE

Marcos Leguizamón*

Tener una pequeña computadora lo suficientemente poderosa y con los instrumentos de navegación necesarios era una parte importante de la misión para viajar a la Luna, pero aun así restaba algo fundamental: el software. El programa que corría la computadora debía ser capaz de controlar todos los dispositivos conectados a ella, tenía que incluir todos los algoritmos matemáticos que describieran los fenómenos físicos a los que se vería expuesta la nave y además debía poder ejecutar todas las maniobras necesarias para controlar el desplazamiento en las distintas etapas del viaje. Este software, o más precisamente firmware porque no se podía modificar luego de salir de la fábrica, era el eslabón final para hacer que todo el sistema de navegación se integrara exitosamente con la nave.

Además de las tareas que debía cumplir para controlar la navegación, el software tenía que operar interactuando con los astronautas continuamente y esto resultaba un desafío importante para el diseño de la interfaz. Hoy es lo más habitual interactuar con una computadora y podemos, por ejemplo, escribir un comando en el teclado y cuando la computadora reconoce la instrucción la ejecuta. Durante el tiempo que la computadora lleva a cabo la acción que le pedimos, que puede ser un segundo, realiza cientos de millones de operaciones, pero durante los varios segundos que se tardó en escribir el comando en el teclado la computadora no tuvo necesidad de realizar tantas operaciones, sino que se limitó a leer las teclas que se presionaban: unas pocas por segundo como mucho. La mayor parte del tiempo la computadora fue subutilizada esperando al humano, que es mucho más lento. Esto no es un problema en la actualidad por el bajo costo de las computadoras pero en la década de 1960 la situación era opuesta: estos equipos eran tan caros y escasos que resultaba ridículo desperdiciar la mayor parte de su tiempo esperando al operador. La práctica habitual era escribir programas o secuencias de comandos en tarjetas perforadas, para lo cual no era necesario una computadora sino una terminal mecánica dedicada a esto. Cuando se tenían todas las tarjetas listas se las ponía en la computadora para que ella las procesara lo más velozmente posible y al terminar la ejecución ya había otra persona

* RTI Latina S.R.L., leguizamon@gmail.com

esperando con más tarjetas. La computadora en las naves Apolo, en cambio, estaba ahí para responder inmediatamente a los requerimientos de los tripulantes y esta comunicación debía ser muy fluida porque eran muchas las operaciones que se realizaban a través de este equipo. El diseño y la implementación de esta forma de interacción requería apartarse de lo que eran las prácticas habituales en aquel momento.

El MIT debía encarar el desarrollo de un software muy complejo que requería cumplir con muchas funciones muy variadas, en una época en donde no existía la experiencia en desarrollo de programas grandes. Lo habitual era que cuando se necesitaba solucionar un problema se escribía un programa para ese fin específico. En este caso el programa debía realizar muchas funciones muy distintas entre sí y varias de ellas a la vez, y, si bien se podía separar en programas distintos, todos debían coexistir en un mismo software. Este software debía escribirse para que entrara en una memoria ROM relativamente pequeña (72 kB) en comparación a la cantidad de tareas que debía realizar y por eso era necesario optimizarlo lo más posible para que pudiera caber ahí. Además, debía aprovecharse al máximo la poca memoria RAM y para lograrlo era necesario estudiar detenidamente cada rutina para usar la menor cantidad posible de variables. Al momento de escribir el programa y estudiar los algoritmos a utilizar también se debía tener en cuenta la poca velocidad de ejecución de esta computadora, que, junto con las otras limitaciones, hacía necesario un análisis detallado de cada rutina para que todo entrara en la memoria y pudiera ejecutarse con la velocidad necesaria.

DESARROLLO DEL SOFTWARE

Desde el comienzo del proyecto, a medida que se diseñaba, fabricaba y probaba el hardware se iban escribiendo programas pequeños para ir poniendo a prueba la computadora. No obstante, los encargados de crear estos programas no estaban dedicados solamente a esta tarea, sino que los escribían como parte de su trabajo en el diseño de la computadora, o en el estudio de cómo realizar determinada operación por medio de esta. Durante los primeros tres años en el MIT no había ningún grupo dedicado exclusivamente al desarrollo del software ni estaba proyectado cómo se crearía el firmware completo de la máquina [1]. La palabra software era una novedad por esos años: no existía mucha experiencia en diseño de programas tan complejos y se subestimaba bastante la complejidad de esta tarea porque era considerado algo secundario. Todas las empresas que trabajaban en distintos componentes para el programa espacial contaban desde el comienzo con múltiples especificaciones de la NASA para realizar el trabajo, así, por ejemplo, las varias etapas del cohete eran diseñadas y fabricadas por distintas empresas pero todas cumplían los requisitos de dimensiones y peso para que luego pudieran encajar y volar juntas, y lo mismo la computadora, que estaba limitada a un volumen, peso y consumo eléctrico determinado entre otros parámetros que debía cumplir. En el caso del software no se contaba con ninguna especificación de cómo debería cumplir con los requerimientos del

sistema de navegación, sino que se iría estudiando a medida que se avanzara en el proyecto y todo quedaría en manos del MIT.

El programa Apolo era un esfuerzo gigantesco y en un momento llegaron a haber 400.000 personas trabajando directamente en él, a través de cientos de empresas, universidades u organismos estatales. Un porcentaje muy chico de esa gente se encargaría del sistema de navegación, pero aun así era un número grande de trabajadores que había que coordinar. El MIT venía de una experiencia muy exitosa con el desarrollo del sistema de guiado del misil Polaris y uno de sus puntos fuertes había sido la coordinación de un número muy grande de empleados, la mayoría contratados de forma temporal durante la etapa que requería mayor trabajo. Esa misma forma de organizar el trabajo les resultó útil para el desarrollo del hardware de la computadora, en donde llegaron a haber en 1965, unas 600 personas dedicadas a esta tarea solamente en el MIT, sin contar el personal dedicado a esto mismo en otras empresas, como en el fabricante Raytheon o los múltiples proveedores de partes. A medida que se fue asentando el diseño del hardware, se fue reduciendo la cantidad de personal dedicada a esta tarea, pero con una computadora lista, ya se debía aumentar el trabajo dedicado a la programación de este equipo.

La idea inicial con la que se empezó el proyecto, era la de asignar las tareas de programación a los diversos ingenieros, matemáticos y científicos que trabajaban en el diseño del sistema de navegación. La lógica detrás de esta decisión era que este personal tan capaz y con gran conocimiento matemático podía aprender a programar con facilidad para aplicar sus conocimientos en un software, pero un programador no podría aprender todos los aspectos científicos y técnicos de la navegación espacial específicos de esta nave con la misma facilidad. Tener un grupo de ingenieros expertos en las naves y otro grupo experto en programación resultaba poco eficiente porque deberían interactuar mucho entre sí para poder llevar a cabo un programa y lo ideal parecía ser saltearse a los intermediarios: en este caso los programadores.

Mientras se escribían los pequeños programas de prueba durante los primeros años del proyecto, esta forma de trabajar resultaba útil y así se iba estudiando cómo realizar cada operación con la computadora, a la vez que se escribían las especificaciones de lo que debería realizar el software de la computadora. Cuando llegó el momento de comenzar el desarrollo del software completo, el que tendría todas las funciones integradas, fue que se hizo notorio que muchas de las suposiciones previas no eran válidas. Para cada etapa de este proyecto se necesitaba gente muy talentosa que encontrara soluciones ingeniosas a los múltiples inconvenientes que surgían, pero la gente muy talentosa en áreas de ingeniería no solía destacarse de la misma manera en la programación. Los programas debían escribirse con mucha habilidad para aprovechar los escasos recursos de la computadora y no alcanzaba con alguien apenas competente sino que se necesitaba sacarle el máximo provecho al equipo. Un experto en trayectorias orbitales podía ser capaz de escribir un programa en un lenguaje de alto nivel para solucionar un problema en una computadora mainframe, pero en este caso el programa, además de obtener la solución correcta, debía hacerlo en la menor cantidad de pasos posibles y no debía interferir con otros programas que usaban el resto de

la memoria. Al final se debió recurrir a programadores dedicados que se encargarían del desarrollo del software siguiendo las especificaciones cada vez más precisas.

La cantidad de problemas que surgían en el desarrollo era muy grande: los desarrolladores luchaban principalmente contra la falta de memoria, tanto ROM como RAM, porque no podían agregar todo lo que necesitaban y eso hacía que debieran buscar técnicas para escribir los programas para ser más pequeños, lo que a su vez producía, en muchos casos, programas con muchos bugs y difíciles de entender. Otro aspecto que se había subestimado durante la planificación era las pruebas que se debían realizar al software. Resultó ser mucho mayor a lo esperado el tiempo que se debía dedicar para verificar que un programa cumpliera de forma correcta su objetivo o encontrar qué bugs existían para luego corregirlos. Como no se avanzaba a la velocidad esperada, se contrataban más programadores y para 1965, ya había unas 100 personas dedicadas al software. Esta cantidad siguió creciendo hasta llegar a los 400 empleados en el año 1968, cuando ya se desarrollaban los firmwares para las primeras misiones tripuladas. Cabe destacar que el desarrollo, con las herramientas y la experiencia que se disponía, era un trabajo muy intensivo que requería mucha mano de obra y mucho del tiempo se iba en tareas repetitivas que hoy serían automáticas, pero de todas formas era muy complejo organizar un grupo tan grande para generar un programa coherente y sin errores importantes. Cada vez que se agregaba más gente al proyecto también había que dedicar más tiempo a entrenar al nuevo personal y la coordinación entre todos los programadores se iba complicando.

Incluso con una gran cantidad de gente dedicada al desarrollo de este software los plazos planificados no se estaban cumpliendo y esto hizo que se comenzara a trabajar muchas horas extras e incluso durante los fines de semana. Esta situación permanecería así durante años [2]. Para 1966 se hizo evidente que el desarrollo del software no estaba avanzando a la misma velocidad que el resto del proyecto y ya estaba claro que la computadora no estaría lista para cumplir los cronogramas establecidos. En la actualidad se tiene una noción mucho mayor del rol del software en un proyecto y ya no llama la atención que un emprendimiento multimillonario se atrase a causa del software, pero esto era una novedad inaceptable en la época. Parecía inaudito que un proyecto de esa magnitud sufriera un retraso por una actividad, a los ojos de los administradores, tan insignificante como era la creación de un programa de computadora.

Cuando en la NASA tomaron conciencia de esta situación, decidieron involucrarse más en el desarrollo del software del MIT, que hasta el momento había sido bastante independiente, entre otras razones por la gran distancia geográfica entre el MIT, en Boston, y la central de la NASA en Houston, Texas. Le asignaron un supervisor, Bill Tindall, un ingeniero que venía de trabajar en los programas espaciales Mercury y Gemini de la NASA, especializado en mecánica orbital. Tindall, que sería el encargado de poner las cosas en orden, fue muy crítico con la forma de trabajo que observó en el área de software e inmediatamente comenzó a aplicar una mayor disciplina. Llevó al MIT a los encargados de desarrollar el software que se usaba en Houston para monitorear a las naves, e incluso al encargado de IBM que ayudó a llevar adelante ese proyecto, para que explicaran su

experiencia en la creación de un software muy complejo. Tindall tenía por costumbre escribir largos memos, conocidos informalmente como *Tindallgrams*, que criticaban fuertemente la forma de trabajar del MIT, lo que produjo situaciones tensas con la gente que trabajaba ahí, pero al poco tiempo esta nueva forma de trabajar empezó a dar sus frutos. Se estudió el programa de forma más general, se identificaron muchas partes que estaban repetidas en distintas secciones y se simplificó la estructura del software, haciéndolo más eficiente. Se implementaron técnicas de control del código escrito, inspecciones visuales de código impreso que fueron muy efectivas para encontrar bugs antes de compilar el programa y ponerlo a prueba. También se mejoró la forma de documentar el trabajo para que los distintos programadores trabajaran de forma más coordinada entre sí.

La misión Apolo 1 iba a ser la primera misión tripulada que utilizaría una computadora modelo Block, y el firmware debía estar listo antes del lanzamiento. Como este vuelo solamente se desarrollaría en la órbita terrestre y con apenas el Módulo de Comando, se usarían muy pocos de los programas en desarrollo para una misión completa hacia la Luna. A pesar de ser una versión más reducida del software, estaba llena de errores que eran conocidos por el MIT pero que no tuvieron tiempo de corregir. La tripulación debería operar la computadora con mucho cuidado para evitar estos defectos [3]. En febrero de 1967 durante una prueba previa varias semanas antes del lanzamiento, los tres tripulantes murieron en un incendio que ocurrió en el Módulo de Control. Esta tragedia desnudó muchas falencias en el programa, desde la coordinación con los distintos contratistas y problemas con el control de calidad. El programa Apolo se paralizó durante meses y se realizaron muchas investigaciones en las que participó el Congreso de Estados Unidos y se reevaluaron muchos aspectos del desarrollo de las naves espaciales. Durante todo ese tiempo la mirada del público estuvo enfocada sobre las naves y los problemas que llevaron a esa tragedia, lo que les dio un tiempo extra a los desarrolladores de software para llegar a completar una versión funcional del software antes del siguiente lanzamiento.

Con la reorganización ocurrida luego de la tragedia, la NASA cambió la forma de trabajo, y el programa Apolo pasó a depender más de la gente encargada del control de misión y los astronautas, y dejó de ser controlada por los encargados de los cohetes que seguían utilizando los mismos métodos desde que se dedicaban a los misiles balísticos. Todas las misiones se retrasaron varios meses y eso les dio tiempo al MIT para ponerse al día con el desarrollo del firmware para la computadora. Al reanudarse los lanzamientos ya se contaba con un software funcional listo para usarse en vuelo. El software nunca llegó a ser un factor que retrasara los vuelos, como sí lo fue el desarrollo del Módulo Lunar en donde se tardó mucho más de lo previsto para tener una nave que se pudiera poner a prueba y hubo que reordenar las misiones de prueba por esta causa.

Versiones del software

El trabajo de los desarrolladores no era crear un solo software sino que debería haber varias versiones distintas divididas, antes que nada, por una característica muy básica:

si eran para la computadora Block I o la Block II. El segundo modelo de computadora tenía un set de instrucciones mucho mayor, era más veloz y los periféricos los manejaba de forma diferente, por eso eran programas distintos los que se debían escribir, aunque los de la Block I eran más simples porque esta computadora no tenía un gran control sobre la nave.

La otra diferencia en las versiones de software estaba dada por la nave en que correría el programa: el Módulo de Comando o el Módulo Lunar. Ambas naves tenían mucho en común a nivel del sistema de navegación, la computadora era la misma, la interfaz de usuario también y lo mismo la unidad inercial, pero las diferencias también eran importantes. Por ejemplo, a pesar que ambas naves se controlaban de forma similar, con propulsores de giro y un motor principal, estos motores estaban dispuestos de forma distinta, y el Módulo Lunar tenía un motor de descenso y otro de ascenso cuando se deshacía de la mitad inferior de la nave, en cambio, el Módulo de Comando sólo tenía el motor del Módulo de Servicio. Además ambas naves tenían distinto peso y distribución de masa, lo cual era muy importante para el software de navegación. Varios periféricos estaban presentes en una nave pero no en otra, por ejemplo el radar de superficie en el Módulo Lunar, el sextante sólo estaba en el Módulo de Comando, en cambio en el Módulo Lunar también se podía usar la computadora para orientar a la unidad inercial en base a las estrellas pero con un telescopio mucho más simple. La diferencia principal entre los programas de cada nave era que existían maniobras que eran propias de ella: el Módulo Lunar debía poder alunizar, lo cual era una tarea muy compleja para el software, y luego debía ascender desde la Luna para reencontrarse con el Módulo de Comando y el Módulo de Comando tenía una maniobra muy distinta al resto que era la reentrada en la atmósfera, en donde la nave no se movía en el vacío ni ganaba velocidad con un motor, sino que estaba constantemente frenando por el rozamiento del aire.

La tercera diferencia en las versiones de software venía dada por el tipo de misión: tripulada o no tripulada. En una misión no tripulada sería necesario incluir operaciones automáticas para que se pudieran controlar desde la Tierra con mayor facilidad al no haber un astronauta supervisando la operación de la computadora.

En el desarrollo se debía tener en cuenta que las misiones espaciales se irían realizando de forma incremental: primero una misión probaba una nave en órbita terrestre, otra misión pondría a prueba la otra nave, más adelante se harían las pruebas ya en órbita lunar, y así hasta realizar una misión completa. Por eso se debía dar prioridad a las maniobras que se realizarían en las misiones más cercanas y se dejaba para más adelante el resto, pero con mucho cuidado de dejar espacio en la memoria para que, llegado el momento, puedan existir todas las funciones en una sola versión de firmware.

La computadora Block I sólo estaba disponible para los primeros Módulos de Comando, por eso sólo haría falta escribir software para esta computadora para esta nave en particular. Se iban a hacer pruebas no tripuladas y también tripuladas con el Módulo de Comando y la computadora Block I, por eso hacían falta versiones del firmware para los dos tipos de misión, pero al final nunca se usó el Block I en misiones tripuladas luego de la tragedia de la Apolo 1.

Cada firmware que se generaba se conocía como *soga* (rope en inglés) por el tipo de memoria ROM que lo almacenaba y al encargado de supervisar la creación de una versión de firmware se lo llamaba *rope mother*. Para identificar a cada versión de firmware (o soga) inicialmente se recurrió a nombres, en general, relacionados al Sol, por Apolo que es el Dios del Sol [1]. La primera fue *Eclipse* que era una prueba básica de teclado y visor de la Block I y que ya incluía el sistema operativo, el *Ejecutivo*, que sin muchos cambios sería el que se utilizaría en el resto de las versiones. La *Sunrise* (amanecer) ya incluía control de la unidad inercial y telemetría. *Corona* era la primera versión pensada para usarse un una misión no tripulada en el Módulo de Comando. Este firmware ya poseía un programa para realizar la reentrada a la atmósfera, aunque era distinto al que usarían los astronautas porque una parte del trabajo la realizaba un equipo electrónico que viaja en la nave en lugar de los tripulantes. *Solarium* era una versión que permitía realizar una misión no tripulada pero con una trayectoria más larga, para las misiones Apolo 4 y 6. *Sunspot* (mancha solar) era la primera versión que habrían utilizado los astronautas de la Apolo 1, todavía en una computadora Block I.

Luego, ya para la computadora Block II, se escribió la versión *Retread* que era una versión muy básica similar a *Eclipse*, y que sólo servía para poner a prueba la computadora. A partir de aquí se comienza a desarrollar versiones de firmware también para el Módulo Lunar y la primera es *Aurora*, que es tiene funciones de guiado básicas y luego *Sunburst* (rayos de Sol) que es la primera versión para utilizar en el espacio, pero en una misión no tripulada y en órbita alrededor de la Tierra. *Sundance* (danza solar) es la primera versión para el Módulo Lunar que se usa de forma tripulada, en la Apolo 9, pero en órbita terrestre, aunque ya tenía programas para utilizar alrededor de la Luna. Para el Módulo de Comando la primera versión para vuelo tripulado de la computadora Block II sería *Sundisk* que se usaría en la Apolo 7 en órbita terrestre pero que ya era capaz de utilizarse en un viaje a la Luna.

Luego de tener cerca de diez versiones todas con nombres distintos y sin un orden aparente, quedó claro que esta forma de identificar a las versiones no ayudaba mucho, y se pasó a una nomenclatura más útil: todas las versiones para el Módulo de Comando se llamarían *Colossus* y todas las del Módulo Lunar *Luminary*, (la C y L inicial era por Comando y Lunar) seguido de un número de versión y, a veces, una letra. Todas estas versiones serían solamente para la computadora Block II. La primera versión en viajar a la Luna sería *Colossus I* en la Apolo 8. En la Apolo 9 se usaría por primera vez ambas naves, la del Módulo de Comando con el *Colossus IA* y el Módulo Lunar con el *Sundance*. A la Luna viajarían ambas naves en la Apolo 10 usando *Colossus II* y *Luminary I* pero sin el software completo para poder descender a la superficie. La primera misión completa, Apolo 11, se llevaría adelante con los firmwares *Colossus IIA* y *Luminary IA*.

Una vez que se tenía una versión final del firmware, este debería ser el mismo para el resto de las misiones y las características propias de cada una (peso de las naves, cantidad de combustible, parámetros orbitales) deberían ir en la RAM. En la práctica, exceptuando las primeras dos pruebas que fueron no tripuladas, cada misión uso una versión mejorada del

software (o firmware) de la misión anterior, aunque ya en las últimas las modificaciones eran correcciones pequeñas.

Programación en lenguaje ensamblador

Las instrucciones que ejecuta la computadora son números, que incluyen el código de instrucción y el parámetro, y el programa entero es, finalmente, una serie de 36.000 palabras de 15 bits. Manejar todos estos números es algo natural para la computadora y, aunque un programador puede escribir un programa generando la secuencia de números adecuada, se hace muy engorroso cuando se tiene un programa muy grande. Para simplificar el trabajo de los programadores se utiliza el lenguaje ensamblador que es una representación en texto de los códigos numéricos que necesita la computadora para funcionar. A cada instrucción del procesador se le asigna un nombre, un mnemónico que en unas pocas letras identifica a qué instrucción se refiere. En distintas arquitecturas se utilizan distintos nombres para representar las instrucciones: por ejemplo en los micros Intel se suele usar la instrucción MOV para copiar de un registro a memoria o de memoria a un registro, en cambio en Motorola se utiliza LOAD (o sólo la L) para cargar un registro desde una dirección y ST, por store, para guardar el valor del registro en memoria. Los mnemónicos que representan a las instrucciones en este lenguaje son más raros: por ejemplo CA (por Clear and Add) es el nombre de la instrucción que copia un valor de la memoria al registro A y TS (Transfer to Storage) es la instrucción inversa, la que copia de A hacia la memoria [4]. La instrucción de salto en las distintas arquitecturas suelen tener un nombre relacionado a *jump* (saltar), por ejemplo JMP, pero en esta máquina se llama TC porque hace referencia a la transferencia del control.

La otra importante ayuda que le brinda el lenguaje ensamblador al programador es la asignación de nombres a direcciones de memoria. En el programa las instrucciones suelen apuntar a otras partes de la memoria, por ejemplo a una variable en RAM, un dato en ROM o un salto a otra parte del programa. Para realizar todas estas operaciones es necesario conocer la dirección de memoria donde están ubicadas las variables o las rutinas que hay que llamar. Por ejemplo, el programador puede saber que la variable con la coordenada X está en la dirección de memoria 563 y la coordenada Y en 232, y entonces cada vez que accede a una de estas variables escribe en el programa 563 o 232. Aparte de lo incómodo que resulta, un problema que surge aquí es cuando se reordena la memoria y, por ejemplo, se reubican X e Y en 344 y 345. En este caso se debería revisar todo el programa y cambiar cada referencia a 563 por 344 y de 232 por 345, y si se usaran 100 veces en el programa habría que cambiarlo 100 veces. El lenguaje ensamblador permite asignarle un nombre a cada dirección de memoria, entonces para usar las variables X e Y simplemente se escribe el nombre X o Y, y es el ensamblador el que resuelve en qué dirección está cada variable y de esta forma el programa es más sencillo de escribir y de leer. Lo mismo se aplica a las direcciones de memoria del programa en sí: desde el listado en ensamblador se le asigna un nombre a una rutina, por ejemplo CALIBRAR, y luego cuando se desee llamar a la rutina

CALIBRAR simplemente se escribe la instrucción de salto seguida de este nombre y el ensamblador la reemplazará por la dirección donde está ubicada.

El ensamblador también genera instrucciones que no son instrucciones reales del procesador sino casos especiales. Por ejemplo, *AD dir* suma el contenido de la dirección dir en el acumulador. Si dir es la dirección donde está mapeado el acumulador (A), se termina sumando A sobre A, por eso existe la pseudoinstrucción DOUBLE que multiplica A por 2, y es lo mismo que escribir en el ensamblador *AD A*.

Las ventajas de escribir un programa en lenguaje ensamblador son muchas con respecto a escribir directo en lenguaje de máquina: el programador se puede abstraer de muchos detalles técnicos al no tener que conocer la ubicación exacta de cada elemento del programa ni memorizar el código numérico de las instrucciones. Pero aun así, la escritura del programa no dispone de muchas ventajas de lenguajes de más alto nivel. Si tomamos como ejemplo esta ecuación matemática:

$$X = \frac{B}{C} + \frac{D\,E}{F}$$

En lenguaje C, lenguaje de alto nivel, se puede escribir de esta forma simple:

```
X = B/C + (D*E/F);
```

En lenguaje ensamblador de la computadora de navegación, utilizando variables de 15 bits, se puede escribir esta fórmula así:

CA	B	carga en A (el acumulador) el contenido de B
TS	L	pasa a L (registro del procesador) el valor en A
CA	7	carga en A el valor en la dirección 7, que es siempre 0
EXTEND		indica que la próxima instrucción es extendida
DV	C	divide los 30 bits formados por A y L por la variable C
CA	L	pasa a A el resultado que está en L
TS	TMP0	guarda el resultado (A) en TMP0
CA	D	carga el contenido de la variable D en A
EXTEND		la próxima instrucción es extendida
MP	E	multiplica A por E. El resultado de 30 bits queda en A y L
EXTEND		la próxima instrucción es extendida
DV	F	divide A y L por la variable F
CA	L	carga en A el registro L (los 15 bits más bajos del resultado)
AD	TMP0	suma a A el resultado parcial que está en TMP0
TS	X	guarda el resultado final (A) en X

Como se puede ver, antes de utilizar la instrucción MP (multiplicación) y DV (división), se debe ejecutar la instrucción EXTEND, que le dice al procesador que la próxima instrucción es extendida y entonces, los 3 bits que indican que tipo de operación es, se deben interpretar de otra forma. La ideal sería que el ensamblador agregara automáticamente la instrucción EXTEND antes de usar una instrucción extendida como MP, pero en este caso es el programador el responsable de indicarlo. La ventaja de hacerlo así es que es más fácil ver cuánto ocupa el programa a medida que se escribe, porque cada línea de código representa una palabra en ROM.

Un factor que hace más complicado a este programa es que en las instrucciones de multiplicación y división se usan los registros A y L para representar valores de 30 bits: en la multiplicación como resultado de multiplicar dos cantidades de 15 bits y en la división como valor a dividir por un valor de 15 bits, pero en ambos casos en A están los 15 bits superiores y en L los bits más bajos. Se necesitan instrucciones extras para mover de L a A cuando es necesario realizar una operación con el resultado que quedó en L o para cargar un valor de 30 bits partiendo del contenido de A. Otra complejidad de este código es la existencia de la variable TMP0 que no está presente en la ecuación original. Es necesario utilizar una variable temporal para almacenar el resultado parcial de la evaluación del primer término de la expresión matemática. En un lenguaje de alto nivel esta variable es un paso intermedio pero aquí es necesario definirla de forma explícita.

Todas las variables que se utilizan (X, B, C, D, E, F y TMP0) son direcciones de memoria que pueden estar tanto en RAM como en ROM, excepto X y TMP0 que sólo pueden estar en RAM porque son modificables. Como este procesador sólo puede direccionar 12 bits por instrucción, los bits restantes se obtienen, al momento de ejecutarse en programa, del valor que tengan los registros EBANK y FBANK, que son los que manejan el mapeo de memoria. Esto significa que antes de ejecutar esta rutina se debe configurar el mapeo de memoria para que todas estas variables se puedan acceder y ese trabajo lo debe realizar el programador escribiendo las instrucciones específicas. Si no todas las variables se pudieran acceder a la vez, entonces en la mitad del cálculo habría que insertar instrucciones para cambiar de bancos y poder direccionar cada variable llegado el momento. El ensamblador solamente toma la dirección del parámetro y la codifica con los 12 bits más bajos porque es un detalle que se tiene que resolver en tiempo de ejecución y es responsabilidad de los programadores. L y la dirección 7 también son direcciones de memoria pero se pueden usar siempre sin ningún cuidado porque están en un área que es siempre visible desde el procesador. Como se ve, los programadores deben estar muy atentos a muchos detalles muy básicos mientras escriben el programa. El ensamblador realiza apenas las tareas más básicas de conversión del código fuente al código ejecutable.

Administración de memoria segmentada

Los sistemas en donde existe más memoria que la que puede manejar el procesador en una instrucción, necesitan usar algún esquema de segmentación de memoria y exigen

mayor trabajo por parte del programador. Por ejemplo, el microprocesador 8086 utiliza direcciones de 16 bits y, entonces, cada instrucción sólo puede trabajar dentro de un bloque de 64 kB. Un programador puede poseer suficiente conocimiento de este microprocesador para escribir un programa que manipule menos de 64 kB, pero si el programa crece y debe utilizar más memoria, ya de sea de datos como de programa en sí, debe recurrir a técnicas propias de esa arquitectura para separar la información en segmentos y dedicar esfuerzo a verificar que ningún segmento supere el máximo de 64 kB de información y a la vez es importante organizar los segmentos según cómo se los utilizará en el programa.

Siguiendo con el ejemplo de una PC con 8086, un programa cuenta con dos tipos de direcciones para apuntar a la memoria: las cercanas (near) y las lejanas (far). Una dirección cercana identifica una posición sólo dentro de un segmento de 64 kB y por eso es de 16 bits. El otro tipo de dirección, dirección lejana, indica no sólo la dirección dentro de un segmento sino el inicio del segmento, que en el 8086 se indica de a múltiplos de 16 bytes, por lo que se necesita un puntero de 32 bits, 16 para indicar el segmento y 16 para la dirección dentro del segmento. Cuando desde un segmento se debe apuntar a información o saltar a una rutina que está en el mismo segmento, se puede utilizar una dirección cercana, pero cuando deba utilizar algo ubicado en otro segmento se requiere utilizar una dirección lejana, que ocupa más memoria y también es más lenta para ejecutarse. Todo lo que debe realizar el programa para acceder a un segmento distinto de el que está usando, ocupa tiempo de procesamiento y hace más voluminoso al programa entero, por eso es muy importante minimizar los cambios de segmentos. Por ejemplo, si se tienen dos variables con las coordenadas X e Y conviene ubicarlas a ambas en el mismo segmento de memoria, en caso contrario, como por lo general se utilizan juntas, luego de leer X se necesitará realizar la costosa operación de cambio de segmento para leer Y. Lo de organizar en segmentos e indicar si las direcciones eran near y far fue una práctica habitual en el desarrollo de programas grandes para PC, e incluso se solía definir lo que se llamaba *modelo* de manejo de memoria, que ayudaba a organizar al programa en segmentos, hasta que apareció el microprocesador 80386 con instrucciones de 32 bits que tiene disponible el manejo de cualquier dirección dentro de un rango de hasta 4 GB de memoria, lo que le permite a cada programa manejar la memoria de forma directa.

Técnicas similares a estas se deben utilizar en la escritura de programas en la computadora de navegación, a pesar de que la arquitectura es muy distinta comparada con el 8086, pero sigue siendo el mismo principio: dividir la memoria en bloques pequeños para que el procesador los pueda utilizar y seleccionar un bloque por vez, evitando cambiar muy seguido esta selección. En este caso, en cada instrucción se puede direccionar sólo 12 bits (4096 direcciones distintas) y se cuentan con cuatro áreas: la memoria RAM fija (768 palabras), la ROM fija (2048 palabras), la ventana a RAM (256 palabras) y la ventana a ROM (1024 palabras). Las ventanas de RAM y de ROM son direcciones donde el procesador puede seleccionar a que parte de la memoria total (38.000 direcciones) accede y son similares a los segmentos de la PC aunque mucho más chicos. El programador debe organizar la información en RAM de forma tal de separar las variables en bloques de 256 direcciones,

tratando de agrupar las variables que se usan juntas para minimizar la cantidad de cambios de ventana necesarios para manipular los datos, y lo mismo con el programa en ROM, que se tiene que separar en bloques de 1024 palabras y, para que cada bloque requiera la menor cantidad de cambios en la ventana de ROM, es ideal que cada rutina llame mayormente a rutinas que están dentro del mismo bloque. Para cambiar una de las dos ventanas se necesita cargar un valor en A y luego guardar el valor de A en el registro EBANK (RAM) o FBANK (ROM) o, para saltar a una parte lejana del código, realizar un cambio de registro Z (puntero de instrucción) junto con FBANK y, si bien no es mucho código extra, puede ocupar una parte muy importante del programa y el tiempo de ejecución de la máquina si no se administra bien la memoria.

Una diferencia de esta computadora con la PC 8086 es la existencia de las dos áreas de memoria fija, una de RAM y otra de ROM. Estas direcciones representan 2816 palabras de las 4096 visibles por el procesador y siempre apuntan a la misma parte de la memoria y esto significa que no es necesario realizar ningún cambio de ventana para utilizar direcciones en estos rangos. Estas áreas de memoria fijas se vuelven muy valiosas por la facilidad de utilizarlas y, como obviamente no se puede poner todas las variables y rutinas en esas áreas de memoria, se debe ubicar allí las variables y rutinas más usadas a lo largo del programa, y se debe administrar con gran criterio.

Numeración octal

Décadas atrás era común para representar información binaria el uso de la numeración octal (base 8), con solamente 8 posibles valores en cada dígito (del 0 al 7). Por ejemplo el número 49 (en sistema decimal) se escribe en octal como 61_8, que corresponde a 6 x 8 + 1. En esta numeración luego del número 7 le sigue, no el 8, sino el 10_8 (1 x 8 + 0). Actualmente, para representar valores internos de una computadora, se suele utilizar el sistema hexadecimal (base 16), que usa por dígito más valores que el decimal (del 0 al 9, y las letras de la A a la F que representan valores del 10 al 15). El hexadecimal es fácil de usar para representar bytes porque utiliza dos dígitos (de 00_{16} a FF_{16}) y cada uno toma 4 bits (un nibble). En el sistema octal cada dígito corresponde a sólo 3 bits y para representar un byte se necesitan 3 dígitos (de 000_8 a 377_8).

Durante el desarrollo se utilizaba mayormente el octal, por ejemplo para las direcciones de memoria o los canales de entrada salida. En el uso de la interfaz de la computadora, cuando se tenía que ver información binaria de la máquina o se debía ingresar algún código, muchas veces se lo hacía usando el sistema octal. Por ejemplo la tabla de estrellas, que asigna un código a cada estrella usada por el software para alinear a la unidad inercial, usa un valor que pasa de la 7_8 (estrella Menkar) a la 10_8 (Mirfak) porque está en octal, pero a este último valor se lo menciona como 10 (diez) a pesar que en realidad es un 8 (ocho).

Interprete de lenguaje de alto nivel

Programar en lenguaje ensamblador es mucho más práctico que hacerlo directo en lenguaje de máquina, pero aun así se está limitado a realizar en cada instrucción una de las pocas operaciones que puede realizar el procesador. Las únicas operaciones matemáticas que posee este procesador son las más básicas operaciones aritméticas con números enteros (suma, resta, multiplicación y división), y para realizar cualquier otra operación más avanzada, por ejemplo para trabajar con vectores, matrices, raíces cuadradas o funciones trigonométricas, se requiere escribir programas que realicen estas operaciones utilizando las instrucciones simples de la máquina. Obviamente, cada vez que se necesita multiplicar dos vectores no es práctico escribir una secuencia que realice esta tarea, sino que lo ideal es tener una rutina que realice una multiplicación de este tipo y, cada vez que se tengan dos vectores que sea necesario multiplicar, simplemente se llama a esta función.

El inconveniente que presenta esta forma de programar es que, a pesar de que se utilizan subrutinas, cada llamada a una rutina requiere cierta cantidad de instrucciones para indicar a que dirección llamar y también para pasarle los parámetros que indiquen sobre qué variables trabajar y donde ubicar los resultados. Si se van a escribir muchas expresiones matemáticas que realicen operaciones avanzadas entonces se ocupará mucha de la escasa memoria ROM de la computadora en pasos básicos para cada llamada a una rutina matemática y, además, se hace muy complicado el programa al tener que escribir tantas cosas similares y así aumenta el riesgo de cometer errores. La solución a este desperdicio de memoria fue la utilización de un lenguaje de alto nivel interpretado. Un lenguaje interpretado es un lenguaje de computadora en donde las instrucciones son ejecutadas, no directamente por la computadora, sino por un programa: el intérprete. La ventaja de un lenguaje de este tipo es que, al estar todo el lenguaje definido en software y no por la electrónica de la computadora, puede ser mucho más flexible y poderoso. Por ejemplo, al tener una instrucción para calcular la raíz cuadrada se evitan todos los pasos necesarios para llamar a la rutina que hace esta operación: aquí es apenas una instrucción simple que ocupa tan solo unos pocos bits en la memoria. El nombre de este intérprete y su lenguaje es simplemente *Intérprete* y es básicamente un programa que está incluido en la misma ROM: cuando se quiere ejecutar algo en este lenguaje, se llama al intérprete con la dirección donde está el programa a interpretar y dentro del programa interpretado se termina con una instrucción que le indica al interprete que vuelva a la ejecución normal.

Máquina virtual

Un lenguaje interpretado implementa lo que se conoce como máquina virtual: una arquitectura de computadora con instrucciones, registros y un procesador que, a diferencia de una máquina real, está implementada en software. Los códigos de instrucciones se los llaman P-Code y, como no tienen las limitaciones de la electrónica, en este caso son mucho más flexibles. Los P-Code en esta máquina virtual son de 7 bits, lo que permite hasta 128

operaciones distintas, por lo cual se pueden tener instrucciones muy específicas que en esta arquitectura se aprovechan para operaciones matemáticas avanzadas: funciones trigonométricas, operaciones entre vectores, matrices o números de 45 bits de precisión (formados por 3 palabras), por poner algunos ejemplos [5]. Al usar 7 bits para la operación, en lugar de sólo 3, se permiten más operaciones pero se limita el espacio para el parámetro de la instrucción, que en lenguaje de máquina es de 12 bits y aquí sólo entrarían 8 si la instrucción ocupara una palabra de 15 bits. Lo que se hace es utilizar otra palabra completa de 15 bits para el parámetro y, de esta forma, se utiliza más memoria por instrucción pero cada una es más poderosa porque puede direccionar gran parte de la memoria disponible en la computadora.

Para ahorrar todavía más memoria lo que se hace es agrupar las instrucciones de a dos, entonces en una palabra de 15 bits entran dos instrucciones de 7 bits. Para acelerar la operación de guardar dos valores de 7 bits en una palabra con un procesador que no tiene instrucciones para girar bits se utiliza una dirección de memoria que gira la palabra que se le guarde ahí 7 posiciones y así existe una parte pequeña del hardware dedicada a ayudar a este software. A continuación se guardan los dos parámetros de las dos instrucciones anteriores: en tres palabras se codificaron dos instrucciones con sus parámetros en un formato un tanto confuso porque mezcla las instrucciones y parámetros en distinto orden pero sirve para aprovechar mejor la escasa ROM. Se ahorra incluso más memoria porque sólo se agregan los parámetros de las instrucciones que los necesitan y así se puede tener dos instrucciones sin parámetros codificadas en una sola palabra de memoria. Cuando se usa un parámetro es siempre una dirección a memoria pero no importa qué banco esté activo porque el intérprete lo resuelve automáticamente, permitiendo al programador obviar los detalles molestos del mapeo de memoria. El parámetro indica cualquier dirección de un rango de 32.000 palabras (RAM y ROM) y el intérprete configura EBANK y FBANK para que se pueda acceder esa dirección. Sólo importa al programador el manejo del bit SUPERBANK para poder seleccionar los 4 kB de palabras de ROM no direccionables con 15 bits.

Otra característica de esta máquina virtual es que tiene disponibles estructuras avanzadas que no estaban disponibles en hardware: pila de llamadas y pila de valores. La pila de llamadas permite realizar una llamada a una función sin tener que guardar, en cada caso específico, el lugar de regreso porque el intérprete automáticamente recupera la dirección desde donde se la llamó a la función y continúa la ejecución donde estaba cuando esta termina.

La pila de valores es una estructura de pila donde se guardan y recuperan números, y que en la máquina virtual reemplaza al registro acumulador en las operaciones matemáticas. Hay instrucciones que agregan un valor en la pila e instrucciones que sacan el último valor que se agregó. Una instrucción que realiza una operación matemática toma el valor que está en el tope de la pila, el último que se puso, realiza una operación matemática y agrega el resultado a la pila. La pila permite ejecutar expresiones matemáticas complejas de forma simple al ir acumulándose distintos niveles a medida que se van resolviendo las distintas partes de la expresión. Por ejemplo, si tenemos esta fórmula matemática:

$$X = \sqrt{\frac{B}{C} + \frac{D\,E}{F}}$$

Una secuencia de instrucciones de la máquina virtual para realizar el cálculo podría ser la siguiente, utilizando números de doble precisión (30 bits) en donde los parámetros de las instrucciones son variables en memoria y cada una ocupa dos palabras de 15 bits:

```
1  DLOAD   B    Carga B en la pila.
2  DP      C    Divide el tope de la pila por C y reemplaza por el resultado.
3  DLOAD   D    Carga D en la pila.
4  DMP     E    Multiplica tope de pila por E y reemplaza por resultado.
5  DP      F    Divide tope de pila por F y reemplaza por resultado.
6  DAD          Suma los dos valores superiores de la pila y los reemplaza.
7  SQRT         Realiza raíz cuadrada del tope de la pila y lo reemplaza.
8  DSTORE  X    Saca el valor en el tope de la pila y lo guarda en X.
```

Este programa es mucho más simple de escribir y leer que el equivalente en lenguaje ensamblador. Una diferencia importa es que ya no es necesario crear una variable temporal para guardar el acumulador mientras se evalúa otra parte de la expresión, sino que al tener los resultados intermedios en distintos niveles de la pila se simplifica esta operación. En el paso 1 la pila queda con un valor de 30 bits y en el 2, este valor es reemplazado por el resultado de la división. En el paso 3 se carga un nuevo valor pero no se pierde el resultado de B/C, sino que queda en la pila pendiente de ser utilizado. En 4 y 5 se realiza sobre el tope de la pila dos operaciones matemáticas hasta obtener el resultado de (D*E)/F. Antes de ejecutar 6, en la pila hay dos elementos: el tope es el resultado del segundo término de la expresión dentro de la raíz y abajo se encuentra el resultado del otro término. La instrucción DAD es similar a DMP en que toma el valor en el tope de la pila y le realiza una operación matemática (una suma en este caso) con su parámetro y el resultado lo guarda en la pila. Como aquí DAD no tiene parámetro lo que hace es tomar el valor en el tope de la pila para usarlo con ese fin y por eso termina tomando dos valores y los reemplaza por el resultado de la suma. El paso 7, una vez que quedó resuelta la expresión dentro de la raíz, realiza la raíz cuadrada con el valor en el tope de la pila y sin ningún parámetro, porque es una instrucción unaria que trabaja sobre un solo valor y reemplaza el tope de la pila por el resultado. El paso final, 8, es tomar lo que quedó en la pila, que es el resultado de toda la expresión, guardándola en una variable en la RAM para que se pueda usar en cualquier otra parte del programa.

En este ejemplo se usaron instrucciones de doble precisión, pero también hay otro conjunto de instrucciones de simple y triple precisión y además instrucciones similares para manejo de vectores, y la escritura es similar en todos los casos con nombres de instrucción ligeramente distintos. Todas las variables que se usan aquí pueden estar en cualquier parte

de la memoria y no es responsabilidad del programador administrar el mapeo de memoria porque de ese trabajo se encarga el intérprete, por eso B y C pueden requerir un cambio de banco para poder acceder y no es problema al momento de escribir el programa.

La enorme desventaja de un lenguaje interpretado es que es muy lento, porque el intérprete debe realizar muchas operaciones para ejecutar cada instrucción. El solo hecho de buscar la siguiente instrucción, decodificarla y ejecutarla, sumado a la administración del mapeo de memoria de los parámetros, termina consumiendo muchas instrucciones del procesador para hacer algo que la computadora hace automáticamente pero en este caso se lo hace por software. Como esta computadora de por sí es muy lenta, la ejecución de programas interpretados demora mucho y el tiempo de ejecución de cada instrucción se mide en milisegundos: las instrucciones simples tardan aproximadamente medio milisegundo, una división de doble precisión unos 2 ms, el producto cruzado de dos vectores 5 ms y las trigonométricas entre 5 y 10 ms cada una, para dar algunos ejemplos.

En base a sus ventajas y limitaciones al decidir qué parte del software se escribe en lenguaje de máquina y que parte en lenguaje interpretado se debe tomar en cuenta todo el espacio en ROM que se ahorra al escribir operaciones matemáticas complejas y también lo lento que será su ejecución.

Sintaxis de lenguaje de alto nivel

Tratándose de un lenguaje de alto nivel, lo lógico sería pensar que la sintaxis también lo es y se pueden escribir expresiones matemáticas para que el compilador las convierta en instrucciones de la máquina virtual. La realidad es que la sintaxis del código fuente de este lenguaje interpretado es similar a la del lenguaje ensamblador que lo único que realiza es reemplazar el nombre de la instrucción por su código. En este caso no sólo el programador debe escribir las instrucciones de la máquina virtual una por una, sino que debe tomarse el trabajo de agruparlas de la forma en que se van a almacenar en la memoria de la computadora: dos operaciones por palabra seguida del parámetro de la primera instrucción y luego del de la segunda. Esta forma intercalada de almacenar las instrucciones en la memoria debe utilizarse también en el código fuente, en vez de dejar que el compilador haga este trabajo.

El ejemplo anterior se debe escribir de esta forma:

```
DLOAD    DP     instrucciones para cargar B y dividir por C
         B      parámetro de DLOAD
         C      parámetro de DP
DLOAD    DMP    instrucciones para cargar D y multiplicar por E
         D      parámetro de DLOAD
         E      parámetro de DMP
DP       DAD    instrucciones para dividir por F y sumar los elementos de la pila
         F      parámetro de DP
```

```
SQRT            instrucción sin parámetros.
DSTORE          una instrucción sola en esta palabra
          X     parámetro de DSTORE
```

Es mucho más complicado escribir así el programa pero se simplifica el compilador y se hace más entendible el volcado de memoria del programa generado, porque cada línea representa una palabra de la memoria. En el ejemplo anterior se ve claramente cómo el programa ocupa 11 palabras en ROM, porque es la cantidad de líneas necesarias para escribirlo

SISTEMA OPERATIVO

El sistema se puede dividir fácilmente entre lo que es hardware y software, pero a su vez el software se puede subdividir en distintas capas, en este caso conviene hacerlo en dos: sistema operativo y aplicaciones. El sistema operativo es un software que provee a las aplicaciones de funciones básicas pero que no están disponibles directamente en el hardware. Por ejemplo en una PC con DOS, el hardware da acceso a un disco con operaciones de lectura y escritura sobre sus pistas, pero el DOS utiliza esa capacidad de control del disco para implementar un sistema de archivos, entonces la aplicación por medio del DOS escribe y lee archivos indicándolos por nombre y dejando al sistema operativo los detalles de cómo organizar toda esa información de directorios y clusters. El programa de aplicación trabaja sobre el sistema operativo y, desde el punto de vista del programador que escribe la aplicación, se puede considerar que el programa corre sobre una máquina extendida que dispone de mayores funciones que el hardware real.

En el caso de la computadora de las naves Apolo, el sistema operativo se llamaba el *Ejecutivo* y, al no existir ni un disco ni ningún medio de almacenamiento secundario, no había necesidad de tener un sistema de archivos, que es uno de los usos principales de un sistema operativo, pero para lo que sí era necesario era para coordinar la ejecución de los programas y para administrar la memoria RAM que le corresponde a cada uno [6]. El Ejecutivo fue escrito por Hal Laning y era un programa muy simple pero poderoso con características muy avanzadas para la época. A pesar de haber sido escrito al principio del proyecto, luego no hizo falta realizarle modificaciones importantes. Durante el desarrollo del software de navegación ya estaba disponible, siempre funcionó bien y no presentó problemas a los programadores. Además, previó muchos inconvenientes que iban a ocurrir.

En un sistema operativo con manejo de archivos, como DOS o UNIX, un programa es un archivo con un nombre (por ejemplo PROGRAMA.EXE) que el sistema operativo le asigna memoria RAM, lo carga y lo ejecuta. En esta computadora todos los programas están en la memoria ROM, igual que el sistema operativo, y se ejecutan directo desde ahí. Los programas aquí se identifican con un número y cuando se le pide al sistema operativo ejecutar uno en particular obtiene los datos a ejecutar de una tabla en ROM con los

atributos de cada programa, principalmente la dirección de inicio para comenzar a ejecutarlo.

Multitarea

El sistema requería realizar muchas operaciones distintas, por ejemplo leer el teclado esperando comandos, controlar los motores y actualizar la velocidad y posición en base al campo gravitatorio que influía sobre la nave en cada momento. Se podía escribir un programa que realizara todo esto a la vez o sino escribir distintos programas que realizaran cada uno una sola tarea y dejar que el sistema operativo los coordinara. Esta última alternativa era mucho más práctica a la hora del desarrollo porque permitía dividir el trabajo mejor al dejar que cada desarrollador se concentrara en un programa que hiciera una sola cosa. El sistema operativo debía, entonces, dar la posibilidad de que varios programas se ejecutaran a la vez para que cada uno se ocupara de su parte. Para que en una computadora que posee un solo procesador se pueda correr más de un programa a la vez, es necesario que el sistema operativo ejecute uno solo en cada instante pero periódicamente salte a otro de los programas activos y así ejecute un poco de cada uno. El sistema operativo se encarga de almacenar la información de la RAM y el estado del procesador de cada programa para que cuando corra lo haga como si fuera el único y cuando pasa el control a otro programa, actualiza esta información para que el otro programa corra como si tuviera el control total de la máquina. Si cada programa se ejecuta durante unos milisegundos da la sensación de que todos se están ejecutando simultáneamente y esto se conoce como multitarea y básicamente existen dos formas de realizarlo, la simple (cooperativa) o la completa (apropiativa o preemptive).

La multitarea cooperativa consiste en que cada programa, mientras se ejecuta, periódicamente llama al sistema operativo para decirle que puede ejecutar otra tarea, si es que hay alguna, y que luego le devuelva el control cuando pueda. El sistema operativo cada vez que recibe esta indicación decide si le pasa el control a otro programa en espera. El programador es el encargado de decidir en qué partes se debe llamar al sistema operativo y lo debe hacer con criterio, buscando que el programa nunca permanezca en ejecución mucho tiempo sin pasarle el control a otro. La contra de este tipo de multitarea es que, si por alguna razón uno de los programas en ejecución se retrasara y no llamara tan seguido al sistema operativo, todos los demás programas también se retrasarían, y si un programa no llama nunca al sistema operativo todo el sistema se cuelga. Si un programa requiere realizar una operación en un lapso de 100 ms pero hay otro en ejecución que llama al sistema operativo cada 120 ms, entonces deberá esperar 120 ms en lugar de 100 exactos. Si, por el contrario, se llamara muy seguido al sistema operativo para permitir cambiar de programa con mayor precisión, se desperdiciaría mucho tiempo del procesador en estas llamadas.

En la multitarea apropiativa el sistema operativo tiene mayor control de los procesos que se están ejecutando, puede interrumpir en cualquier momento a cada uno para pasarle el control a otro, y eso permite que ningún programa se quede con el 100% de la ejecución.

También simplifica en muchos aspectos la escritura de los programas porque ya no es necesario insertar periódicamente una llamada al sistema operativo sino que este cortará automáticamente la ejecución cuando llegue el momento. El timing es más preciso porque se puede hacer que un programa tome el control cada, por ejemplo, 100 ms exactos sin importar que esté realizando el programa en ejecución. Este último tipo de multitarea es la forma más poderosa de administrar varios procesos concurrentes y su uso tiene muchas ventajas pero también presenta muchos inconvenientes a la hora de implementarlo. Se necesita una disciplina importante en la escritura de cada aplicación al acceder a recursos (RAM, periféricos) compartidos con otro programa porque puede ocurrir que se pase el control al otro programa mientras no terminó de modificar un recurso en común, y lo que se hizo de forma incompleta puede dejar al recurso inutilizable para ambos. El scheduler, la parte del sistema operativo dedicada a administrar el tiempo de ejecución de los programas, debe estar muy pulido porque es una tarea muy compleja la que debe realizar. Un sistema corriendo varias tareas de esta forma no es determinista, esto significa que no se puede predecir cómo se va a comportar, sobre todo cuando ocurren conflictos entre dos programas que corren a la vez, porque una falla puede ocurrir una vez pero luego esa condición de error es probable que no se vuelva a repetir en miles de pruebas y esos errores difíciles de reproducir también son muy complicadas de corregir. La estrategia en este caso es prevenir estos problemas por medio de gran disciplina al momento de diseñar y escribir los programas.

En las computadoras personales tardó mucho en aparecer la multitarea. En la PC, el DOS no permitió nunca ejecutar más de un programa por vez, apenas permitía dejar un programa *residente* y que este se ocupara (accediendo directamente al hardware) de tomar el control de la máquina periódicamente. Recién el Windows 1.0 en 1985 incluyó multitarea, pero sólo a partir del Windows NT en 1993 para máquinas profesionales y Windows 95 en 1995 para equipos hogareños se comenzó a usar multitarea completa en lugar de sólo cooperativa. En las Macintosh el sistema operativo Mac OS desde el principio tuvo multitarea pero recién en el 2002 con el Os X se adoptó el uso de multitarea completa.

El Ejecutivo trabajaba con multitarea cooperativa porque estaba lejos de sus posibilidades implementar un scheduler complejo, y además se complicaría mucho la escritura de programas que pueden ser cortados en cualquier momento. Cada programa debe llamar periódicamente a una función, llamada NEWJOB, que se encargaba de pasarle el control a otro programa si era necesario. Al escribir cada programa se debía verificar que se llamara cada aproximadamente 20 ms, estimando el tiempo de ejecución de cada sección de código y con cada llamada debían suponer que a continuación los registros habían sido modificados, porque el sistema operativo no se tomaba el trabajo de restaurar estos valores. Los programas interpretados llamaban a esta función en cada instrucción, así se ahorraba tener que escribir una instrucción específica para esto, pero se desperdiciaba tiempo.

Ejecución por nivel de prioridad

La multitarea en este sistema operativo trabaja con prioridades [6]. Cada programa tiene un nivel de prioridad y el sistema operativo debe garantizar que se ejecute siempre el de mayor prioridad y que el resto queden en espera. Un programa de prioridad alta puede ser el que controla el motor de la nave que tiene que encenderlo y apagarlo en un momento específico. Uno de prioridad baja puede ser el que actualiza información en el visor, porque si se actualiza la información una fracción de segundo más tarde no hay consecuencias graves, pero sí las habría si se mantuviera el motor encendido una fracción de segundo más. El nivel de prioridad de cada programa se puede modificar mientras corre, y es necesario hacerlo, así cuando el programa está haciendo algo importante se autoasigna una prioridad alta y cuando simplemente espera que ocurra algo se pasa a una baja prioridad. El sistema operativo entrega el control según la prioridad y es importante que los programadores asignen las prioridades teniendo en cuenta el funcionamiento de todo el sistema, porque si simplemente se indicara que un programa tiene siempre una prioridad alta en lugar de tomarse el trabajo de asignarle la prioridad que le corresponde en cada momento, existiría la posibilidad de que nunca se le entregara el control a otro programa.

Para los casos en que se necesitara un timing preciso, por ejemplo para ejecutar una acción en 100 ms, se puede realizar de una forma más limitada. Se le indica al Ejecutivo que en un lapso de tiempo, con definición de 10 ms porque se utiliza el Timer 3 para este fin, debe ejecutar determinado programa. El sistema operativo mantiene una lista con los programas a ejecutar en base al reloj y configura al Timer 3 de forma tal de que genere una interrupción cuando llegue el momento de ejecutar el próximo programa de esta lista. Estos programas que se ejecutan de esta forma, con timing preciso, cortan la ejecución del programa que esté corriendo en ese momento, pero se ejecutan con las interrupciones deshabilitadas para no permitir a la vez ser cortados por otra interrupción, por eso están limitados en cuánto tiempo pueden trabajar porque si no impedirían el funcionamiento del resto del sistema. Estos programas se usan para funciones con un timing exacto, como en el control de los motores.

Si por algún inconveniente el sistema operativo no puede pasarle el control a otro programa en la lista de espera, por falta de memoria RAM libre o no puede cumplir sus requisitos de tiempos de ejecución, se genera un error que se guarda en una variable en RAM y dispara la rutina de error, que en las naves activa, a su vez, la alarma general que le indica a la tripulación que algo ha ocurrido [7]. El operador debe escribir en la terminal de la computadora el comando para ver el código del último error y recién ahí se entera que fue lo que falló.

Estos errores del sistema operativo estaban pensados principalmente para la etapa de desarrollo de software para detectar problemas, tanto en un programa en particular como en el mismo sistema operativo. Lamentablemente, en la misión Apolo 11 durante el primer alunizaje y en la etapa más delicada se sucedieron una serie de eventos que no

fueron detectados con anticipación y desencadenaron una serie de cinco alarmas de este tipo durante el descenso.

Programas resistentes a los reinicios

Ya avanzado el desarrollo del software surgió un nuevo requerimiento muy importante un año antes del primer alunizaje. En la NASA se decidió que todos los programas críticos deben continuar su ejecución con normalidad luego de un reseteo [2, 6]. Cuando la computadora recibe la señal de reset, el procesador deja de hacer lo que estaba haciendo y comienza a ejecutar desde una posición inicial fija. Normalmente lo que se hace en esta rutina inicial es borrar la RAM y empezar todo de nuevo, pero ahora se requería que en este momento se revisara que programas estaban en ejecución antes de resetearse y se continuara ejecutándolos, de esta forma al producirse un reseteo no se detendría una maniobra en ejecución, ni se continuaría con la maniobra por más tiempo del necesario, lo que podría tener en ambos casos consecuencias muy graves.

En computación normalmente se diferencia entre *hard reset* (también llamado *cold reboot*) cuando se apaga la computadora y se la vuelve a encender pero habiendo perdido la información que estaba en la RAM, y el *soft reset* (o *warm reboot*) que es cuando sólo se reinicia el procesador pero se mantiene la información en RAM. En el caso de esta computadora en particular, como la memoria está implementada con núcleos magnéticos, nunca se pierde la información en RAM con una interrupción del suministro eléctrico, por eso siempre se puede considerar que un reinicio es un *soft reset*. La rutina inicial debe, entonces, verificar el estado de variables especiales de RAM donde se indica que programas estaban en ejecución y se vuelve ejecutar esos programas.

Pero implementar esa funcionalidad no es algo simple que esté en una sola rutina de software sino que hay que modificar todos los programas para que puedan trabajar así. Para escribir un programa que ejecute una serie de pasos, lo más simple es escribir los pasos uno atrás del otro y dejar que el procesador los ejecute en orden. Por ejemplo un programa simple que enciende un motor durante 20 segundos y luego lo apaga podría implementarse así:

1. Encender el motor.
2. Esperar 20 segundos
3. Apagar el motor

El problema de esta implementación es que si se reinicia el programa en el paso 2, habiendo pasado 10 segundos, se vuelve al paso 1, se vuelve a encender el motor (que ya estaba encendido) y se espera otra vez 20 segundos antes de apagarlo. En total se mantuvo encendido el motor 30 segundos, con un resultado totalmente distinto al esperado. Para que el programa pueda continuar la operación sin repetirla desde el inicio se debería implementarlo como una máquina de estado. Por ejemplo se podría hacer así:

1. Leer variable con estado del programa.
2. Si el estado es 0 (motor apagado) encenderlo, guardar la hora en que hay que apagarlo y pasar a estado 1 (esperando). Volver al paso 1.
3. Si el estado es 1 (esperando) verificar si se llegó a la hora del fin de la espera. Si se llegó, apagar el motor y pasar al estado 2 (programa terminado) y terminar la ejecución del programa.

Al reiniciar el programa, se vuelve a la ejecución desde el paso 1 pero se trabaja según el estado de las variables en RAM, de esta forma el programa continúa ejecutando lo mismo que estaba haciendo antes de reiniciarse. Esta forma de escribir el software es más compleja, requiere un mejor diseño, pero termina produciendo programas más robustos. La contra es que ocupa mucha más memoria, tanto ROM para el programa, que tiene más instrucciones, como RAM para guardar el estado de cada programa, por eso por este requerimiento se terminó agrandando la cantidad de RAM y ROM hasta llegar al tamaño final del diseño. No todos los programas requieren implementarse así, porque no es importante que todo se deba continuar en un reinicio, por ejemplo las rutinas que muestran información en el visor. El sistema operativo tiene en una tabla en ROM los atributos de que debe hacer con cada programa al reiniciarse, así según qué programas estaban en ejecución se los reinicia con la rutina adecuada.

Esta funcionalidad del sistema permite que en caso de una falla en el hardware de la computadora no se detenga la ejecución de una maniobra importante del sistema de navegación. Pero también es muy útil para problemas de software, porque si un programa no vital realiza algo que impide la normal ejecución de un programa importante, con resetear la computadora, al descartarse el programa innecesario y recuperar el que estaba haciendo algo crítico, se solucionan problemas que son difíciles de detectar en el momento. Esto resultó muy importante en la misión Apolo 11 donde varios reseteos causados por el sistema operativo no afectaron la maniobra crítica que se estaba realizando.

CICLO DE DESARROLLO

Para tener un software que se pudiera ejecutar en la computadora, había primero que pasar por una serie de pasos que comenzaba por el trabajo de los programadores [1, 2]. Cada programador se encargaba de una parte del software y la escribía en lenguaje ensamblador, en lenguaje interpretado o una combinación de ambos, pero lo hacía a mano en hojas de papel. Luego esas hojas eran pasadas a otros programadores que las estudiaban, buscaban errores y analizaban si esa sección cumpliría con el objetivo para la que fue escrita. Cuando esta parte del software era finalmente aprobada, las hojas escritas a mano eran entregadas a un equipo de operarios, que no necesariamente conocían de programación, y ellos se encargaban de copiar el texto en tarjetas perforadas, utilizando máquinas especializadas para este fin y no computadoras. El resultado era una serie de tarjetas perforadas con el listado del programa en lenguaje ensamblador y lenguaje interpretado.

Hasta este momento ninguna computadora había participado en la creación de ese listado: todo se hizo de forma manual, o con máquinas electromecánicas. Las tarjetas perforadas obtenidas representan el vínculo en común entre los operadores humanos y la computadora.

A estas tarjetas de cartón se le realizaban pequeños agujeros que la computadora, por medio de un lector electromecánico u óptico, podía leer y obtener información según la distribución de estas perforaciones, pero, a la vez, las tarjetas eran relativamente fáciles de leer e interpretar por los operadores humanos. La información que se escribía en las tarjetas era texto, 80 columnas por tarjeta, pero no se lo hacía de forma binaria: no eran combinaciones arbitrarias de bits, sino que por cada columna se marcaban, como mucho, 3 agujeros. En cada columna, con 1 o 2 agujeros a un lado, se indicaba el tipo de código que se usaba (si era un dígito del 0 al 9, una letra de la A a la I o una de la J a la R) y luego se seleccionaba, en un grupo de 10 posibles perforaciones, el valor dentro del rango seleccionado. Las tarjetas eran grandes y poseían espacio para comentarios escritos a mano y también tenían impresa, en cada columna junto a las perforaciones, la letra que representaban, aunque gracias al hecho de tener una codificación simple, era relativamente simple leer su información grabada en las perforaciones por alguien entrenado. Si se quería insertar una línea de texto al programa, se lo podía hacer agregando una tarjeta entre medio de otras dos, pero siempre cuidando de no desordenar la pila de tarjetas que equivale a desordenar el programa.

Cuando se tenían todas las tarjetas, que formaban el código fuente del programa, se las cargaba en la computadora que corría el compilador. Este compilador se llamaba YUL, y era a la vez un ensamblador que convertía las instrucciones de lenguaje ensamblador a instrucciones de la computadora de navegación y un compilador que generaba la secuencia de bits que necesitaba el *intérprete* para ejecutar un programa de alto nivel. El YUL corría inicialmente era una IBM 650, que era una computadora con válvulas de vacío, luego se migró a una Honeywell 800 por unos años y hacia la etapa final del proyecto corría en una IBM 360, que era la principal computadora de uso tanto científico como comercial en la segunda mitad de los ‘60. Como el compilador se ejecutaba en una computadora totalmente distinta a la computadora de destino, el YUL se considera un compilador cruzado.

Como cada tarjeta representa una línea de código fuente y en el YUL, ya sea lenguaje ensamblador para generar código para el procesador o en el lenguaje interpretado para generar código para el intérprete, por la forma en que se estructura las sintaxis de ambos lenguajes cada línea corresponde a una palabra de la ROM. Como el programa de la versión final usará, o estará cerca de usar, la totalidad de la ROM de 36.000 palabras de 15 bits, entonces se necesitarán esa cantidad de tarjetas perforadas. Si se tiene en cuenta que, además de las líneas para generar código, existen textos con directivas para el compilador y comentarios, la cantidad de tarjetas será aun mayor. El solo trabajo de manipular todo ese cartón es una tarea muy ardua para los operadores y a la computadora le lleva mucho tiempo cargar todas las tarjetas y luego debe procesar su contenido y darle varias pasadas para, por ejemplo, resolver los símbolos usados y todo el proceso lleva horas. Lo ideal, para

ahorrar tiempo, sería compilar sólo las partes del código fuente que se modificaron, pero ese es un nivel de sofisticación que no estaba disponible en este compilador. Años más tarde se comenzó a separar a las herramientas de desarrollo en compilador y linkeador, en donde el compilador toma una parte del código fuente y genera un archivo binario intermedio llamado objeto y el linkeador arma el programa completo uniendo todos los objetos que se compilaron por separado y si se modifica un solo objeto no hay que recompilar el resto: alcanza con volver a ejecutar al linkeador. La desventaja de usar archivos objeto intermedios es que hay que organizar el programa de forma tal que se pueda separar en partes y se deben definir muy precisamente que partes del objeto se pueden acceder desde otro objeto y cómo, y, al haber una mayor complejidad, también surgen problemas nuevos y las herramientas deben estar mucho más pulidas para evitar estos inconvenientes. En este caso, con cada mínima modificación al código fuente se tiene que compilar el programa completo y para eso se debe recorrer todo el código fuente. Esta tarea tan lenta se la deja para la noche así el resultado está listo para el siguiente día y se debe organizar todo el trabajo de los desarrolladores para que al final de la jornada haya un código fuente para poder compilar.

Una vez que el YUL procesaba todas las tarjetas, si todo salió bien se obtenía un programa ejecutable y este resultado final la computadora lo volcaba, no sobre tarjetas perforadas, sino sobre una tecnología similar: sobre una cinta perforada. Esta cinta de papel enrollada poseía una serie de perforaciones para codificar la información, todas en una línea, que la hacían más adecuada para el manejo de datos binarios para uso de la computadora, pero no para la manipulación por parte de personas. No era posible interpretar esta serie de agujeros a simple vista, y por eso este rollo sólo servía para cargarlo en otra computadora, pero ya era información binaria: una serie de valores de 16 bits, no texto.

El siguiente paso, una vez que se contaba con el código binario, debería ser crear la memoria ROM de la computadora, pero como el proceso de creación de la memoria de sólo lectura duraba semanas, no sería posible esperar todo ese tiempo para poner a prueba cada versión del software porque el ciclo de desarrollo sería interminable. Lo que se hacía era cargar el resultado final en una memoria de núcleos magnéticos que tenía la misma capacidad y la misma interfaz que la ROM de sogas, pero por supuesto ocupaba mucho más espacio y no sería posible enviarla al espacio. Esta computadora de prueba se conectaba a varios dispositivos similares a los que se usarían en el espacio, como la terminal y el sextante, y el resto de los periféricos, por ejemplo la nave espacial en sí, se simulaban en una computadora más grande con un programa dedicado a esto, pero, como la velocidad no le alcanzaba para simular todo en tiempo real, estaba a su vez conectada a otros equipos conocidos como computadoras analógicas, que eran dispositivos electrónicos que simulaban algunos procesos, por ejemplo detalles de la física del movimiento en órbita, y con la ayuda de estos equipos se podía probar el software como si se estuviera en el espacio.

Existía otra forma de probar el programa y era por medio de un emulador que corría en una de las computadoras grandes del MIT y cargaba todo el firmware en su RAM para simular de forma precisa, ciclo a ciclo la ejecución de la computadora de navegación. Junto

con la simulación de la computadora también se simulaba toda la física de la nave en las distintas etapas de la misión. La contra de usar este simulador es que no podía correr en tiempo real porque era muy complejo todo lo que debía realizar, tanto para emular a la computadora como a la nave y por eso no se operaba a través de un teclado sino que se escribían con anticipación las teclas que se irían presionando a medida que se desarrollara la simulación que tardaba más que la ejecución real. La ventaja de esta simulación es que se tenía un control muy preciso de todo lo que ocurría y se podía analizar cómo se ejecutó el programa en cada ciclo si se imprimía el listado detallado de la ejecución. Esta prueba, además, al ser digital era determinista porque siempre que se ejecutaba producía los mismos resultados a diferencia de las computadoras analógicas que tenían variaciones cada vez que se usaban. La creación de la memoria ROM de sogas solamente se realizaba cuando se requería una versión para probar el sistema completo y cuando se preparaba la misión con una versión ya probada exhaustivamente.

Junto con el binario para grabar en la ROM el YUL generaba, a la vez, un listado impreso con el programa generado para poder analizarlo. Este listado se imprimía en una impresora de alta velocidad en un formulario continuo y, para las versiones finales, eran unas 1700 páginas de textos y números, que en impresión simple faz representaba una pila muy grande de papel (Figura 1). Se imprimía el código fuente completo que se ingresó y junto a cada línea se mostraba en qué posición de memoria ROM se almacenaba su contenido y el valor, en octal, que se obtuvo al compilar esa línea aprovechando que cada línea de código fuente representa una palabra de memoria. Mirando este listado se puede verificar si el código generado es correcto y se puede analizar mejor. Unas 300 páginas del listado corresponden a estadísticas de generación de código y a la lista de símbolos usados: indica qué posición de memoria se asignó a cada nombre de rutina o variable y se muestra en orden alfabético para que se pueda buscar el nombre de un símbolo y revisar dónde se encuentra en la memoria. Con esta información del listado se puede analizar la ejecución de los programas en el emulador sabiendo que parte del programa está corriendo y que variables modifica.

INTERFAZ DE USUARIO

Contándose ya con una computadora, un teclado y visores, todavía no estaba diseñado cuál sería el método en que se interactuarían los astronautas con la computadora. La NASA había llegado a las especificaciones de cuántos visores harían falta según la información que se determinó que se necesitaría ver en cada momento y también se especificó las dimensiones del teclado, pero la secuencia en que se le daría instrucciones a la computadora no había sido definida. No existía mucha experiencia en la época de diseño de interfaces en tiempo real para computadoras, pero, incluso si lo vemos desde el punto de vista actual, sigue siendo algo complicada la tarea porque no se contaba con visores alfabéticos que pudieran mostrar palabras y que permitieran elegir textos con opciones de

un menú describiendo lo que estaba realizando en cada momento: aquí sólo se contaba con teclado y visores numéricos y todo se debía codificar con dígitos.



Figura 1. Margaret Hamilton, supervisora de desarrollo de software en el MIT, al lado de listados generados al correr el programa.

Mientras la NASA no definiera una forma de interactuar con la computadora de todas formas era necesario probar la computadora y las versiones preliminares del programa durante el desarrollo y entonces se necesitaba una forma de interactuar con la computadora de forma temporal. Ramón Alonso ideó un método simple que consistía en ingresar oraciones elementales, como *mostrar posición*, *mostrar hora*, *encender motor* o *alinear sistema inercial*: instrucciones simples basadas en un verbo y un sustantivo, o bien, usando terminología de computadoras más modernas, un comando seguido de un parámetro. Como todo debía ser numérico, lo que se haría sería asignarle un código a cada verbo y otro código

al sustantivo. Sobre esta base, Alonso con Albert Hopkins y Herb Thaler crearon una interfaz temporal que fue la que comenzaron a usar tanto los encargados del sistema como los astronautas que participaban del desarrollo del sistema de navegación [2]. A todos los que la usaron le satisfizo esta forma de comunicarse con la computadora y a los astronautas le resultó práctico, así que cuando llegó el momento de diseñar una interfaz se propuso simplemente seguir utilizando esta solución temporal pero de forma definitiva. Inicialmente se pusieron objeciones a utilizar esta forma novedosa de comunicarse porque no era lo suficientemente científica o militar, pero la terminaron aceptando y fue la que se utilizó en todas las misiones a la Luna (Figura 2).

Secuencia de operación

Los astronautas interactúan con la computadora de forma directa a través de las palancas de mando, cuyas señales son procesadas por el software para dirigir la nave, y también otros controles simples, como los interruptores para indicar el modo del piloto automático. De todas formas, las instrucciones más avanzadas son ingresadas a través de la consola, tanto por los tripulantes de la nave como por los operadores en Tierra que pueden interactuar con la computadora remotamente. Por el teclado se puede consultar el estado de los programas, configurar a la computadora y ejecutar acciones. La rutina del programa que controla el teclado y los visores se llama *Pinball Game* y es una de las primeras cosas que se escribieron para esta computadora.

La secuencia para ingresar un comando por teclado se inicia presionando la tecla VERB [6]. Luego de esto, la computadora queda en espera de que se ingrese un valor del 0 al 99 que identifique a uno de los verbos existentes, y deben ingresarse obligatoriamente dos dígitos. No se puede, por ejemplo, ignorar un cero a la izquierda: si el código del verbo es el 1 debe escribirse 01. Uno de los visores de dos dígitos es exclusivo para el verbo y ahí se puede ver el valor ingresado. Presionando la tecla ENTR se acepta el verbo y la computadora ejecuta la acción requerida. Antes de aceptar con ENTR se puede agregar un parámetro de también dos dígitos a la operación, que en este caso es el sustantivo: se presiona la tecla NOUN (sustantivo) seguida de los dos dígitos del código que lo identifica. El sustantivo también tiene un visor propio dónde se ve el código ingresado y, entonces, se puede confirmar visualmente ambos valores antes de aceptarlos. Presionando ENTR se acepta el verbo y el sustantivo y la computadora ejecuta la acción.

Un grupo de verbos muy usados son los que van del 01 al 06, que hacen que la computadora muestre determinado valor en uno de los 3 visores de 5 dígitos, llamados R1, R2 y R3. Por ejemplo el verbo 05 (mostrar valores octales 1, 2 y 3 en R1, R2 y R3) seguido del sustantivo 9 (código de última alarma) lo que hace es mostrar en el visor el código del último error que ocurrió. El verbo 06 (mostrar valor decimal en 1, 2 o 3 visores según el verbo) seguido de sustantivo 33 (tiempo de ignición) muestra la hora de ignición de la próxima maniobra. Los verbos del 11 al 16 son similares a los del 01 al 06 pero, además de mostrar una variable en uno de los visores, el valor que se muestra en el visor se actualiza cada un

segundo y así se puede monitorear continuamente el estado de una variable. Con distintas secuencias de estos verbos y los sustantivos que representan a las variables existentes se puede configurar los visores para mostrar distintas combinaciones de datos.

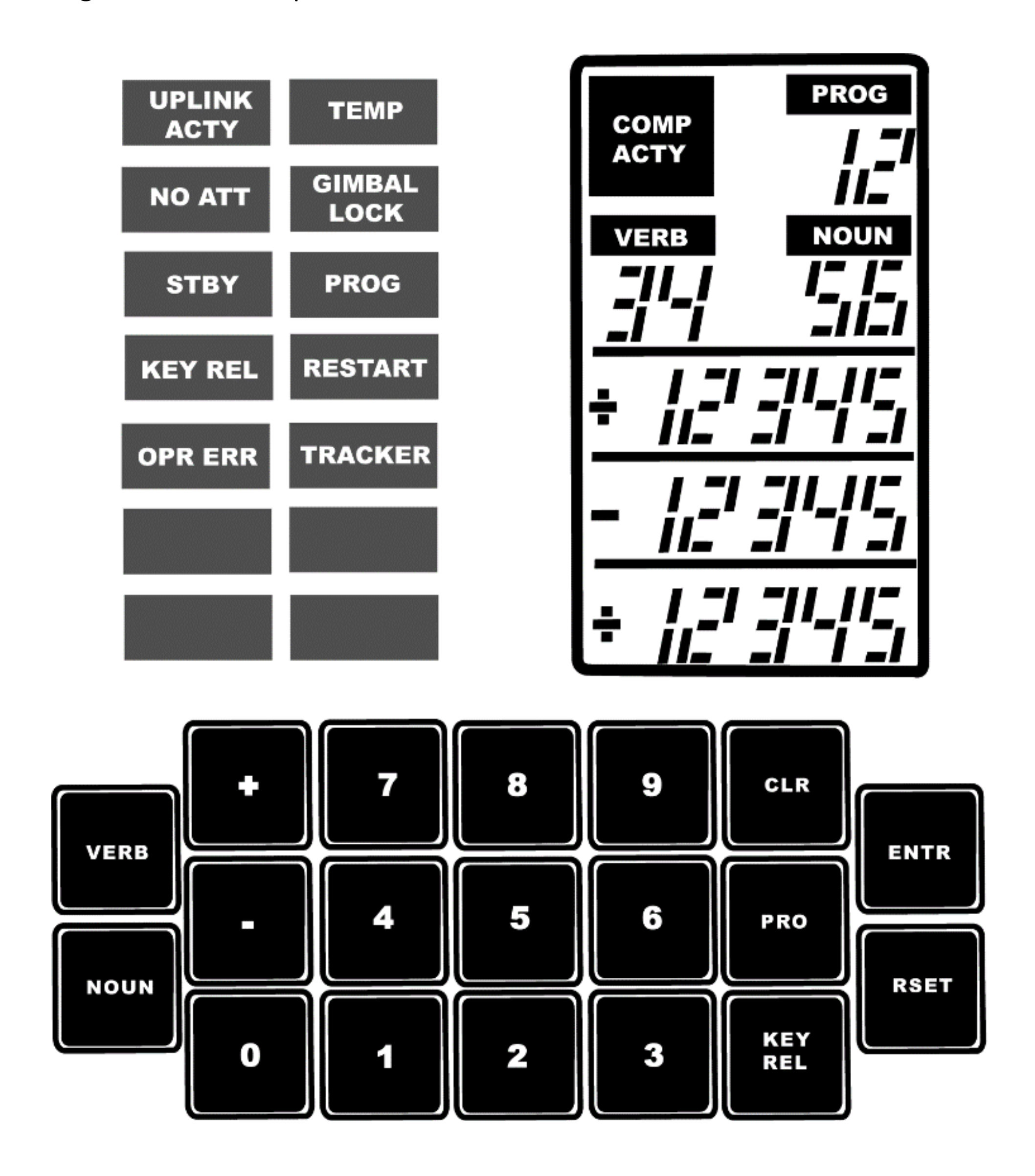


Figura 2. Esquema del DSKY: teclado y visores de la computadora

Muchas acciones requieren más datos, además del verbo y el sustantivo. En estos casos, luego de aceptar la operación con ENTR, la computadora solicita, a través de los visores, el ingreso de valores: los visores de verbo y sustantivo comienzan a parpadear mostrando dos códigos que indican, para un astronauta que conoce esos valores, que es lo que se debe ingresar. Dependiendo de qué dato se necesite, se ingresa un valor que se va mostrando en uno de los tres visores de 5 dígitos. Este valor se puede ingresar usando dos tipos de codificación: si se escriben solamente dígitos numéricos, es un valor expresado en octal (con dígitos del 0 al 7), en cambio, si primero se presiona una de las teclas con signo (+ o -) los siguientes 5 dígitos que se ingresen serán decimales (del 0 al 9). No existe forma

de ingresar en el teclado ni de ver en el visor el punto separador de la parte entera y la parte decimal, por eso los valores son siempre enteros. Si hiciera falta utilizar un valor con decimales, por ejemplo 1,2 millas, tanto el programa como el operador deben multiplicarlo por un factor ya predefinido, por ejemplo 100, entonces se ingresaría y mostraría el valor como 120. Una vez que se ingresó el valor se lo confirma presionando la tecla ENTR o se lo vuelve a escribir presionando CLEAR. Si el comando requiere que se ingrese otro dato, otra vez parpadearán los códigos que identifican al dato en los visores de verbo y sustantivo. Recién cuando se hayan ingresado todos los valores requeridos, se comienza la ejecución del comando.

Esta es la secuencia más común, que permite ingresar comandos y sus parámetros, pero no todo se puede realizar de esta forma y en algunos casos se varía la secuencia. En determinados momentos, un programa necesita mostrar información en el visor para avisar de algo al operador, a pesar de que no se había pedido esa información desde el teclado. El programa puede mostrar esa información en el visor, siempre y cuando este esté libre, y eso es determinado por la tecla KEYREL: una vez que se presiona esta tecla, el visor puede ser actualizado por un programa corriendo en segundo plano. Si no se había habilitado la tecla KEYREL entonces el programa no puede escribir en el visor pero se enciende una luz (*Key Rel*) para avisar que un programa está queriendo utilizar el visor y no puede. Se debe ser cuidadoso al diseñar una interfaz de usuario en que tanto el operador como la computadora pueden tomar control de la terminal en cualquier momento: el operador para ingresar un comando y la computadora para avisar de un evento que ocurrió. Incluso en la actualidad, con interfaces mucho más avanzadas y muchísima experiencia acumulada, existen programas que interrumpen al operador mostrando ventanas pop-up mientras este está en el medio de una operación y hacen difícil retomar el control. En este caso se utiliza una técnica bastante complicada en muchos detalles, pero efectiva una vez que se gana experiencia y evita interrumpir innecesariamente al astronauta en el medio de un ingreso de datos que puede ser muy largo y complejo.

Indicadores de vuelo

Además de los visores numéricos la computadora muestra información en luces que se encuentran a la izquierda del teclado y sirven para indicar algunos estados de forma rápida. Algunas de esas luces se controlan desde el software y otras son independientes. La luz *Comp Acty* indica que la computadora está funcionando y no se halla en el modo stand-by para ahorro de energía, en cuyo caso se enciende la luz *STBY*. La luz *Temp* está conectada a un sensor e indica que la unidad inercial está trabajando fuera del rango aceptable de temperaturas.

Gimbal Lock indica que la plataforma inercial está cerca de entrar en este modo y los astronautas deben maniobrar para evitarlo para no perder su referencia de orientación. *No ATT* avisa que no se está recibiendo información de actitud desde la unidad inercial y *Tracker* que hay un problema con el sextante del Módulo de Comando. El radar de superficie del

Módulo Lunar tiene dos luces que indican que no está recibiendo información: la luz *Alt* permanece encendida mientras no se llegue información de altitud y *Vel* es similar pero para indicar que lo que no se recibe es la velocidad. Si ambas luces están apagadas significa que el radar funciona correctamente.

La luz *Key Rel* avisa que un programa que corre en segundo plano está intentando mostrar algo en el visor pero no puede porque está siendo usado por el operador. Cuando el operador ingresa una secuencia de teclas erróneas se enciende la luz *Opr Err* para avisarle del error. La luz *Prog* avisa que ocurrió un error en la ejecución de un programa. *Restart* indica que se reinició la computadora por algún error que ocurrió. El indicador *Uplink Acty* se enciende mientras llega información de forma remota a la computadora. Esta información consiste en teclas que se están presionando desde la central de control de vuelo y llegan a la computadora por radio.

UNIDADES DE MEDIDA

En los Estados Unidos se usan aún hoy las unidades de medida imperiales (pulgadas, pies, millas, libras, etc.) a diferencia del sistema métrico decimal utilizado casi universalmente. En el caso del proyecto espacial Apolo se optó por el sistema métrico, por su mayor practicidad en la conversión entre unidades, por ser el utilizado en ámbitos científicos y además porque gran parte del personal que diseñaba los cohetes era de origen europeo, sobre todo alemán. El programa de la computadora estaba escrito justamente utilizando estas unidades en cada lugar donde se representa una cantidad física, ya sea de longitud, masa, velocidades o aceleración.

Los astronautas y el personal de control de vuelo, por otro lado, estaban muy familiarizados en su labor diaria con las unidades imperiales y como durante una misión debían recurrir todo el tiempo mediciones de distinto tipo, no resultaba práctico que se adaptaran a un sistema nuevo porque podría ser causa de errores. Lo que se hizo fue que toda la planificación de la misión se realizara con unidades del sistema métrico pero en los instrumentos y en toda la información que manejarían los astronautas se utilizaran las unidades imperiales. En el caso de la computadora de navegación esto significaba que el programa estaba escrito para hacer las cuentas con el sistema métrico, por ejemplo representando la masa de la nave en kilos, la posición en metros y la velocidad en metros por segundo, pero a la hora de mostrar la información a la tripulación la convertiría en una unidad imperial, por ejemplo convirtiendo metros en pies. Lo mismo debía realizarse en el sentido inverso, cuando se ingresara información habría que transformarla del sistema imperial a la unidad del sistema métrico correspondiente.

La tripulación utiliza para la altitud cuando se está cerca de la Tierra o la Luna y para distancias entre dos naves cercanas los pies (1 pie es 30,48 cm). Para posición en el espacio se utilizan millas náuticas (1 milla náutica es 1,85 km) que no hay que confundir con las millas terrestres que es una unidad bastante más pequeña. Para la masa de la nave se utilizan libras (1 libra es 0,453 kg). Nótese que se utilizan para distancia dos unidades

distintas, pie o milla náutica, según el tipo de distancia que se mide y que la conversión entre ambas se hace multiplicando por un factor de varios dígitos (6080) en cambio en el sistema métrico es más simple la conversión entre metros y kilómetros porque sólo hay que mover la coma. Para velocidad no se utiliza nudos (milla náutica/h) sino pies/s, que puede expresar las grandes velocidades de hasta 11 km/s (36089 pies/s) con precisión sin necesidad de utilizar dígitos decimales.

Para los ángulos se utilizan grados con valores de –180 a 180. El problema de esta unidad es que para mostrar este rango de información se contaba con 5 dígitos más el signo: los 3 primeros dígitos eran la parte entera y quedaban dos para decimales (usando sistema decimal, no minutos y segundos). Para la mayoría de los usos esos 2 decimales eran suficiente precisión, pero cuando se mide longitud en un sistema de coordenadas polares, por ejemplo para identificar un punto en la Luna, cada grado no tiene siempre el mismo tamaño en metros en la superficie, entonces en el ecuador lunar 0,01 representa más de 300 metros, y ese es un rango de error muy alto, mucho mayor que la definición del sistema de navegación. Como el problema es el primer dígito que se desperdicia en gran medida (sólo puede ser 0 o 1, los otros 8 valores no se usan) la solución encontrada fue usar un rango distinto para los ángulos de la longitud: de –90 a 90. En este rango (que corresponde a los grados reales divididos por dos) se usan sólo dos dígitos para la parte entera y se cuenta con 3 decimales, lo que da una precisión 5 veces mayor, aunque el operador debe estar atento para distinguir los dos tipos unidades de medición de ángulos.

SISTEMAS DE COORDENADAS

Las coordenadas usadas por sistema de navegación corresponden a un determinado sistema de referencia que define un punto inicial (la coordenada 0, 0, 0) y una orientación para los tres ejes (x, y, z). Es importante tener bien definido el sistema de coordenadas entre todos los que participan en la misión, las naves y los controladores de vuelo en Tierra, para que puedan compartir información sobre posiciones y direcciones en el espacio sin ambigüedades. La unidad inercial de las naves debe estar debidamente alineada con este sistema de referencia, sus tres acelerómetros deben apuntar de forma paralela a los tres ejes seleccionados como referencia para que las aceleraciones medidas correspondan a esos ejes. El software de navegación también debe utilizar el mismo sistema de coordenadas para calcular la posición de la Tierra y la Luna en el espacio y que coincidan con la información de la unidad inercial.

Un sistema de referencia que podría ser útil para usar toda la misión, que transcurre tanto en la Tierra como en la Luna, podría ser tomando al Sol como referencia, el punto inicial sería el centro en el Sol (la coordenada 0,0,0) y el plano x, y podría coincidir con el plano orbital terrestre alrededor del Sol, para el eje x se podría tomar la dirección desde el Sol hacía la Tierra en un día en particular (por ejemplo el 1 de enero a las 12 AM de Greenwich del año en que se realiza la misión), el eje y es perpendicular al x en el plano en que están ambos, y el eje z es perpendicular a los otros dos ejes. De esta forma se tendría un

sistema que con tres números identifica cualquier punto del espacio y podría ser útil para cualquier viaje a otros planetas del Sistema Solar.

Si bien simplificaría utilizar un solo sistema de referencia, existe un inconveniente y es que a una persona le puede resultar complicado interpretar las coordenadas. Por tomar un ejemplo simple, si se tomara una coordenada con respecto al Sol mientras se orbita alrededor de la Tierra, la coordenada en un punto de la órbita podría ser 150.000.000, 0, 0, (en km) y en el otro extremo de la órbita 150.012.000, 0, 0 a pesar estar en un punto opuesto de la Tierra los números no cambian mucho porque sigue siendo más importante en magnitud la distancia al Sol que la posición alrededor del planeta. Además, como la Tierra gira alrededor del Sol (a unos 108.000 km/h), en cada órbita se obtendrían números muy distintos y sería muy difícil separar un movimiento que no nos importa (la Tierra alrededor del Sol) del que sí nos importa (la nave alrededor de la Tierra). Casos como este también ocurrirían si se tomara un sistema de referencia con centro en la Tierra mientras se órbita la Luna o al revés, pero, en cualquier caso, para la computadora no sería un problema interpretar estas coordenadas, en cambio para los astronautas, controladores de vuelo y los programadores del software de navegación sería mucho más entendible usar un sistema de referencia que tenga más que ver con la trayectoria que se está siguiendo en cada momento.

La NASA creó muchos términos para definir los innumerables elementos científicos y tecnológicos relacionados a las misiones Apolo, generalmente siglas y acrónimos y para identificar los sistemas de referencia usados a los largo de las misiones se creó el raro acrónimo REFSMMAT (a partir de Reference to Stable Member Matrix, o sea matriz de referencia del miembro estable). El REFSMMAT indica cómo debe orientarse la unidad inercial (o miembro estable) y define hacia que parte del cielo debe apuntar y también cual es el punto 0 del sistema de coordenadas. Durante una misión se cambiaba en distintas etapas el sistema de referencia (el REFSMMAT) según la trayectoria que se estaba siguiendo. En las cercanías de la Tierra se usaba el centro de esta como centro de coordenadas y para las maniobras alrededor de la Luna se toma su propio su centro. En el lanzamiento la unidad inercial queda alineada hacia arriba en el preciso momento del lanzamiento, así era más fácil evaluar el ascenso y en el alunizaje y la reentrada a la Tierra, en donde se parte desde su órbita a un punto de la superficie del planeta, se define un inicio de coordenadas que pasa por el centro de la Luna o la Tierra, según el caso, y uno de los ejes pasa por el punto calculado para el descenso, así con solo mirar las coordenadas fácilmente se evalúa rápidamente cuanto falta para alcanzar el destino.

Existían hasta ocho sistemas de referencia distintos según la misión y se cambiaba cuando se iba a realizar una maniobra en particular para que mirando los números quedara claro si se cumplió el objetivo buscado, por ejemplo cambiar de plano orbital o ganar velocidad en una dirección en particular. Si se pasaba a un sistema de referencia alineado con la nueva dirección o el nuevo plano en el cual moverse, con ver la nueva velocidad ya estaba claro que se cumplió el objetivo buscado. En cada cambio de sistema de referencia se debe reorientar la unidad inercial para que sus acelerómetros queden paralelos a los nuevos

ejes de coordenadas. La computadora lo hace automáticamente en base a la orientación anterior, girando la unidad inercial hasta coincidir con la nueva dirección. Técnicamente se podría cambiar de sistema de referencia sin reorientar la unidad inercial, que los ejes de los acelerómetros no sean paralelos a los del sistema de coordenadas y luego sus mediciones corregirlas por software, pero implicaría mayor procesamiento para la computadora y mayores desviaciones por sumas de errores en la conversión. También el Gimbal Lock era un tema a tener en cuanta al seleccionar un sistema de referencia, porque se debía tomar un sistema en el que nunca hiciera falta orientarse cerca de la posición de Gimbal Lock. Normalmente, en una maniobra, alcanza con moverse en un plano en particular y rotar la nave también dentro de ese plano, por eso se debe orientar la unidad inercial de forma tal de que este lejos de las orientaciones prohibidas.

SUPERVISIÓN DE OPERACIONES CRÍTICAS

Para guiar a la nave el software debe controlar a la nave, encendiendo y apagando los motores entre otras operaciones con periféricos conectados a la computadora. El problema es que si el astronauta a cargo de la nave no supervisara las acciones que realiza la computadora, en caso de ocurrir un error en el software o en un parámetro cargado por el tripulante, la computadora podría realizar acciones riesgosas para la nave y sus tripulantes. Por ejemplo, para controlar los distintos motores existen interruptores en el panel de la nave que habilitan que la computadora pueda controlarlo. El astronauta debe poner el interruptor, o fusible, en la posición habilitada antes de correr un programa que lo utilizará, sino no hay forma de que la computadora pueda encender ese motor. Lo mismo se debe realizar para los motores que no serán utilizados, aunque en este caso hay que deshabilitarlos para evitar el caso de que un error en el software los active.

Las naves disponen de varias cargas explosivas que se detonan para cortar uniones entre distintas secciones que se van descartando. Por ejemplo para separar al Módulo de Comando del Módulo de Servicio o del cohete Saturno V. Los detonadores de estas cargas explosivas no son controlados por la computadora porque no se confiaba tanto en el software y en la electrónica de la computadora en general como para darle una tarea tan riesgosa. Los tripulantes deben conectar el circuito eléctrico del detonador y manualmente disparar el detonador, con interruptores protegidos para evitar usarse por error.

Los programas que controlan el motor principal de la nave tienen que cumplir un requisito extra además de tener habilitado los interruptores para que la computadora pueda controlar al motor: el software debe consultar al astronauta antes de encender el motor, si el operador no presiona la tecla PRO el programa no puede comenzar a mover a la nave. El software puede tener los algoritmos para controlar el movimiento de la nave pero la última palabra la tiene siempre el tripulante. Los astronautas están muy interiorizados en el funcionamiento de la nave y en el de la computadora en particular y por eso son capaces de supervisar su funcionamiento y controlar que sólo utilice de la nave lo que le corresponda en cada momento.

PILOTO AUTOMÁTICO

Una función muy importante que cumple la computadora es la del piloto automático, que mantiene a la nave orientada en el espacio. Es necesario orientar a la nave de forma precisa antes de ejecutar una maniobra en una dirección en particular o para realizar una observación, pero también es necesario mantener la orientación de la nave una vez que quedó apuntando en la dirección deseada porque existen distintos factores que la hacen rotar. Cada vez que se enciende el motor principal la nave gana velocidad de rotación porque es muy difícil que este esté perfectamente alineado con su centro de masa, por eso es necesario contrarrestar estos movimientos para que la nave quede apuntando en la dirección correcta durante una maniobra. La expulsión de líquido refrigerante al vacío o de desperdicios y también la separación o la unión entre las dos naves producen fuerzas que hacen rotar a la nave y es necesario anular esas rotaciones lo antes posible para que la nave no quede girando de forma descontrolada. También existe el inconveniente del Gimbal Lock en la unidad inercial que hace que la nave tenga prohibida determinadas orientaciones en el espacio, por eso también es necesario evitar rotar sin control para nunca entrar en este estado. Hace falta un mecanismo automático para ocuparse de la tarea de mantener a la nave siempre orientada porque si solamente se pudiera hacerlo manualmente los tripulantes deberían estar continuamente con los controles de los motores atentos a anular cualquier rotación que se produzca antes de que la nave gane velocidad y rote descontroladamente.

Que el software se encargue de mantener orientada la nave permite actuar con gran velocidad y con una gran eficiencia en el consumo de combustible, porque las operaciones están optimizadas para realizarse en la menor cantidad de pasos [1, 8]. La operación básica para orientar la nave en los tres ejes es disparar un par de motores opuestos hasta ganar velocidad en un eje y esperar, viendo la medición del ángulo de la unidad inercial, hasta llegar cerca de la posición deseada y ahí se encienden los motores opuestos a los anteriores para anular la velocidad de rotación en ese eje y luego esto mismo se hacen con los otros dos ejes.

Esta operación no es tan sencilla y hay que resolver unos cuantos detalles para poder llevar a cabo de forma eficiente. Los motores de orientación siempre empujan con la misma potencia pero lo que sí se puede regular es cuánto tiempo permanecen encendidos, porque mientras más tiempo lo hagan más velocidad de rotación se ganará, y estos tiempos son muy pequeños, medidos en milisegundos, por eso el programa debe ser muy exacto para encender cada motor la cantidad de milisegundos que se necesitan sin salirse de un margen de error muy pequeño. El control del tiempo también debe ser preciso para decidir con cuánta anticipación, antes de llegar al ángulo buscado, se deben encender los motores para detener la rotación de la nave en el ángulo correcto. Para calcular estos tiempos el programa necesita una estimación muy aproximada del peso de la nave, que depende de la cantidad de combustible en los tanques y si está unida a la otra nave o si ya se expulsó una etapa,

entre otros detalles, para calcular la velocidad de rotación a ganar según el tiempo de encendido de los motores.

El programa este, además de saber cómo moverse en un eje de rotación, necesita convertir los ejes de referencia de la unidad inercial, que son los que utilizan tanto los tripulantes y el software para guiarse, a los ejes de rotación de los motores, porque estos dos sistemas de referencia no coinciden entre sí. Los ejes de rotación los debe calcular teniendo en cuenta la geometría de nave y distribución de los motores de giro y el centro de masa de la nave, que varía, igual que la masa, con la cantidad de combustible y si está o no unido a la otra nave.

El piloto automático del Módulo de Comando y el del Módulo Lunar cumplen la misma función y se utilizan de igual manera, pero su implementación en gran parte es distinta porque la geometría, centro de gravedad y distribución de motores es distinta en cada nave. Tanto el Módulo de Servicio como el Módulo Lunar tienen 16 propulsores de giro, distribuidos en 4 grupos con 4 motores cada uno, en cambio el Módulo de Comando, una vez que se deshizo del Módulo de Servicio, tiene solamente 12 de estos motores para orientarse.

Además de conocer las características físicas de la nave, el programa, cada vez que realiza una operación de rotación, debe obtener feedback de la propia unidad inercial y determinar así si la nave está girando de la forma en que se le indicó. Si detectara que el movimiento que envío no se está realizando como debería, puede calcular si un motor no está operando con normalidad o está deshabilitado y buscar otra forma de hacer el giro. Como se debe poder rotar en tres ejes y para girar en una dirección en un eje se necesitan dos motores opuestos y se debe poder girar en ambas direcciones, da como resultado que son necesarios 12 propulsores para orientarse en cualquier dirección. En las naves que disponen de 16 se cuenta con motores redundantes que el software puede utilizar en caso de producirse fallas en otro par de motores.

Este programa se ejecuta 10 veces por segundo y cada vez que lo hace tarda, aproximadamente, 25 milisegundos, lo que deja solamente el 75% del tiempo del procesador para realizar todas las demás tareas. Al ejecutarse tan seguido, el piloto automático detecta cualquier pequeña rotación y la contrarresta lo antes posible para mantener a la nave bien orientada todo el tiempo. Se le puede indicar a la computadora desde el teclado la orientación que debe tener la nave y estos ángulos se guardan en variables en la RAM y el piloto automático realiza, en segundo plano, todos los disparos de los motores para rotar en esa dirección y luego para mantenerse apuntando así, contrarrestando otras fuerzas que inciden sobra la nave. Esta orientación en la computadora se puede cambiar de forma mucho más directa por medio de la palanca de rotación. Al mover esta palanca en uno de sus tres ejes se le envía una señal a la computadora y esta utiliza esta información para modificar las variables con la orientación para que luego el piloto automático tome esos ángulos como la dirección hacia donde debe apuntar la nave. Siempre existe la posibilidad de que los tripulantes controlen directamente los motores de giro, pero es muy ineficiente en duración y consumo de combustible porque se debe operar

con mucha prueba y error al no poder realizar los cálculos y estimaciones que efectúa la computadora.

Un parámetro muy importante para configurarle al piloto automático es el llamado *deadband*, que es la tolerancia con la que se mantiene la orientación actual. Existen dos opciones: 0,3 grados o 5 grados. Si el deadband es de 5 grados el piloto automático, cada vez que revisa los ángulos de la unidad inercial, 10 veces por segundo, solamente encenderá los motores para corregir la orientación de la nave cuando vea que en uno de los ejes existe una diferencia de más de 5 grados en relación a la variable en RAM. Mientras más chica sea la tolerancia más se deberán usar los motores para contrarrestar las rotaciones de la nave. Si no se está en una maniobra que requiera de gran exactitud, un deadband de 5 grados permite que la nave no rote descontroladamente y ahorra combustible porque ignora muchas pequeñas rotaciones.

No solamente es necesario orientar la nave con un sistema de referencia fijo en el espacio sino que a veces es necesario dejar a la nave rotando en determinado eje. Cuando se orbita alrededor de la Luna se desea que la nave mantenga su orientación con respecto al suelo, así, por ejemplo, se puede observar el terreno a lo largo de toda la órbita. Para lograr esto se necesita que el piloto automático deje a la nave rotando lentamente, dando una vuelta completa sobre sí misma en el mismo tiempo en que realiza una órbita completa alrededor de la Luna. Cuando se configura al piloto automático en este modo para orbitar, este debe calcular la orientación que necesita la nave en cada punto de la órbita y la velocidad de rotación en que debe poner a la nave para quedar en este estado.

CÁLCULO DE ACELERACIÓN, VELOCIDAD Y POSICIÓN

Cada dos segundos se ejecuta un programa que actualiza la información de aceleración, velocidad y posición de la nave en base a lo leído de los acelerómetros y la posición con respecto a los campos gravitatorios terrestre y lunar. Según la posición en que se encuentre la nave realiza el cálculo de la aceleración de la gravedad en ese punto, tomando la distancia a la Tierra y a la Luna, y esta aceleración obtenida, multiplicada por los dos segundos desde la última medición, se suma a la variable donde está la velocidad actual de la nave. También se le suma a la velocidad la aceleración leída desde los acelerómetros, también integrada a los dos segundos desde la vez anterior en que se leyó. Luego se toma la velocidad de la nave y se la suma a las variables con las coordenadas en el espacio de la nave, antes se sumarla también se multiplica por los dos segundos desde la última vez que se sumó el anterior cálculo.

De esta forma cada dos segundos la computadora cuenta con variables que indican la velocidad de la nave y las coordenadas, según el sistema de coordenadas en uso. El resto de los programas pueden usar esta información para mostrarla en el visor o tomar decisiones de navegación según la maniobra que se esté efectuando.

OBSERVACIONES ASTRONÓMICAS

El sextante del Módulo de Comando permite realizar mediciones ópticas muy precisas y se pueden leer los ángulos de ese instrumento desde la computadora, pero es el software el que debe realizar todos los cálculos para obtener la orientación y posición de la nave en base a esas mediciones. En la ROM se tiene una lista de 38 estrellas, cada una con sus coordenadas en el cielo según un sistema de referencia común a todas. Para seleccionar una estrella se le indica a la computadora el código que tiene esta estrella en la lista usando la numeración octal, por ejemplo el código 12 octal es Rigel, la 14 Canopus, 15 Sirio, 36 Vega y 40 Altair (Figura 3). Si se le indica a la computadora que apunte hacia una de estas estrellas, rotará la nave y al sextante en una orientación tal que permite observar esta estrella desde el visor de este instrumento. Si la calibración de la unidad inercial está bien, al mirar por el sextante se podrá observar esta estrella en el centro.

La computadora que iba a usar el MIT en la sonda a Marte debía realizar la calibración de forma automática, pero en el programa Apolo la computadora cumplía esta tarea con la asistencia de los astronautas. Existen equipos independientes que realizan esta función sin necesidad de control humano llamados star trakers, y los satélites y misiles intercontinentales los suelen utilizar para controlar la actitud orientándose por medio de las estrellas.

Si hiciera falta calibrar a la unidad inercial, se hace rotar a la nave buscando una de las posibles 38 estrellas hasta que se la pueda observar por el visor del sextante y luego se ajusta el sextante hasta que la estrella quede en su centro y se le indica a la computadora que ahora está centrada. El software conoce la orientación de la nave con respecto a la unidad inercial, la orientación del sextante con respecto a la nave y las coordenadas de la estrella hacia la que apunta el sextante, entonces, con esos datos, calcula la desviación de la unidad inercial. Si se realiza esta operación con dos estrellas distintas, la computadora calcula cuanto hay que ajustar la unidad inercial para dejarla bien orientada en las tres dimensiones. En el visor se le muestra este valor al astronauta y, si lo acepta, aplica este cambio sobre la unidad inercial.

El otro uso que se le daba al sextante era para calcular la posición de la nave en el espacio. Para eso, en lugar de simplemente apuntar hacia una estrella, se utiliza la capacidad del sextante para medir ángulos entre dos objetos. Primero se orienta la nave para apuntar hacia el horizonte de la Tierra o de Luna con el visor fijo del sextante. Luego se va moviendo el otro visor del sextante hasta apuntar en su centro a una estrella conocida hasta que el horizonte y la estrella, que el astronauta los ve superpuestos, quedan perfectamente alineados. Como la nave se mueve a una velocidad considerable con respecto a uno de los cuerpos observados, ya sea la Tierra o la Luna, sobre todo teniendo en cuenta que se observa con un aumento de 28X, entonces esta alineación durará un instante, y en ese mismo momento se debe presionar la tecla MARK. La computadora toma la hora en que se hizo la medición y el ángulo en que se encontraba el horizonte y la estrella en ese instante dado. Como también conoce el movimiento orbital de la Tierra y la Luna y sabe la posición

que deberían tener en el momento preciso en que se realizó la medición y también conoce las coordenadas de la estrella observada, con esos datos calcula desde que punto del espacio se realizó la medición y esa es la posición de la nave.

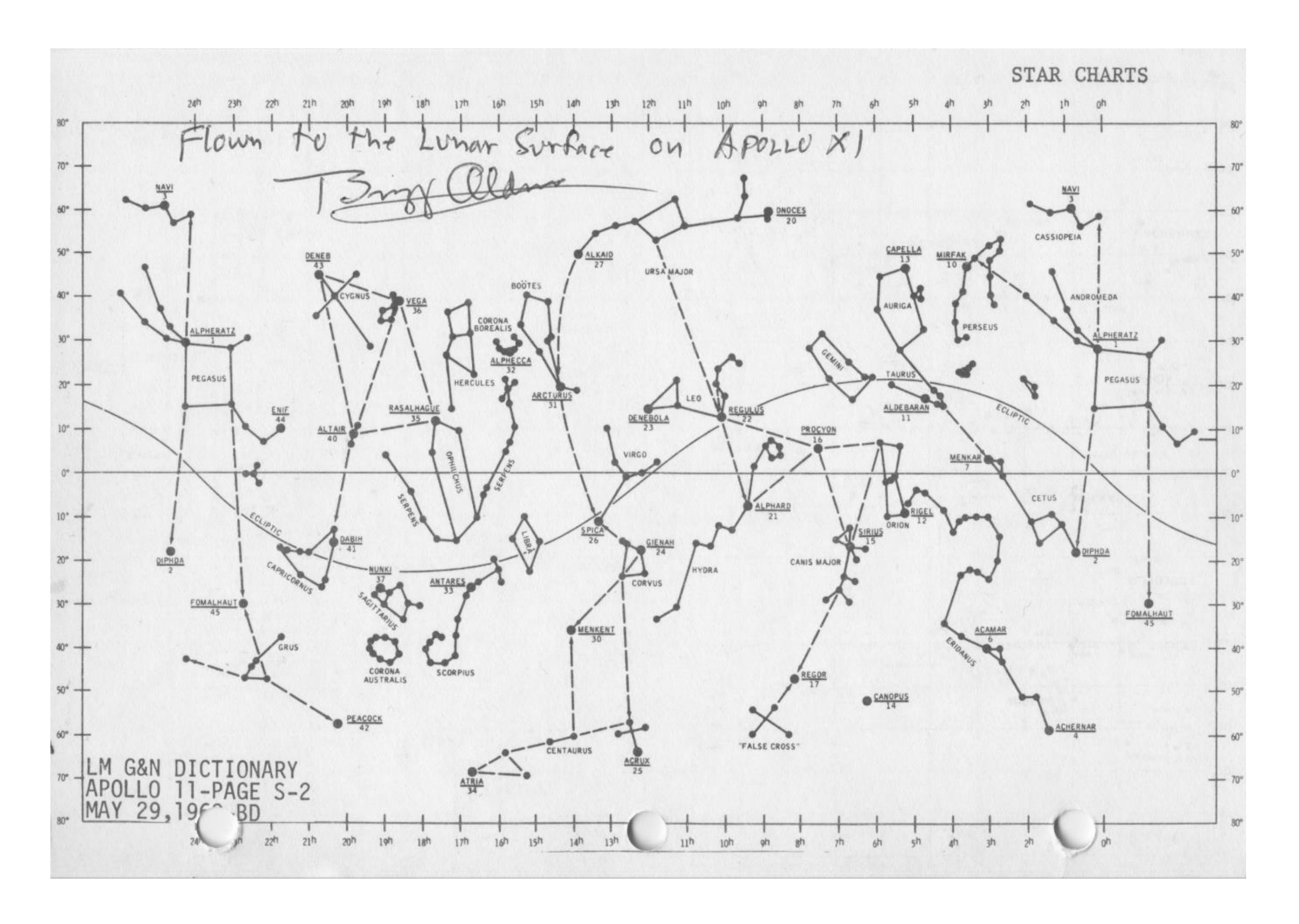


Figura 3. Mapa de constelaciones con los códigos de las estrellas que se usó en la Apolo 11.

Si bien la precisión del sextante es muy alta, cualquier pequeña desviación en las mediciones puede producir que se calcule una posición muy alejada de la real. Para corregir desviaciones en las observaciones el software usa una técnica matemática conocida como filtro de Kalman que se había desarrollado poco tiempo antes de comenzar el programa Apolo y el MIT la estudió exhaustivamente para utilizarla en la navegación. El resultado de una medición se puede tomar como la suma del valor real que se intentó medir y ruido. Este ruido son las desviaciones producidas por imperfecciones en los instrumentos y, si se considera que la distribución de este ruido es completamente al azar, el filtro de Kalman ayuda a filtrarlo en gran medida para obtener mediciones más precisas. Esta es una técnica estadística que trabaja sobre una serie de mediciones, nunca en una aislada, promediando el ruido entre mediciones y utilizando estimaciones de margen de error de cada sensor del sextante. La implementación en el software es bastante sencilla porque con cada medición que se efectúa se utilizan coeficientes para filtrar el ruido y a la vez se actualizan estos valores para que se usen en la próxima medición. No requiere una gran cantidad de procesamiento extra salvo unas pocas operaciones aritméticas por cada medición.

La consecuencia de utilizar este filtro es que las mediciones van mejorando a medida que se hacen más. Las primeras observaciones que se realizan no arrojaran buenos resultados y es necesario efectuar varias hasta que se empiecen a hacer más precisas. Es muy importante que todas las observaciones las haga la misma persona y use siempre el mismo criterio, porque, por ejemplo, en la determinación del horizonte la atmósfera terrestre hace difícil decidir dónde termina la Tierra y empieza el espacio, pero si en todas las mediciones se toma este límite en el mismo lugar los cálculos mejoran.

EJECUCIÓN DE MANIOBRAS

Como se explicó en secciones anteriores, la nave se desplaza en el vacío del espacio por medio de una serie de impulsos, aunque no está constantemente maniobrando, la mayor parte del tiempo los motores permanecen apagados. Por ejemplo, en el trayecto de la Tierra a la Luna, la nave abandona la órbita terrestre con una maniobra (inyección translunar) realizada por el cohete Saturno V que durante 6 minutos le da un impulso que hace que se dirijan hacia la Luna [9]. Si bien la maniobra esta se realiza de forma muy precisa, no se pueden evitar pequeñas desviaciones y en distancias tan grandes estas desviaciones hacen que la nave se aleje mucho de su destino, por eso hay programados a lo largo de todo el trayecto una serie de maniobras de corrección de rumbo a medida que se van midiendo la desviación, así en un viaje de unos tres días se realizan de 3 a 7 maniobras de ajuste de dirección, de unos pocos segundos cada una. Cuando la nave se aproxima a su destino, realiza otra maniobra de varios minutos que la pone a orbitar la Luna. En este ejemplo vemos cómo se realizó un gran trayecto de varios días y cientos de miles de kilómetros por medio de unos pocos impulsos de los motores, pero obviamente son impulsos aplicados de forma precisa en una dirección en particular y calculados para poner a la nave en una trayectoria específica.

La computadora realiza cada una de las maniobras de la nave, conoce que a determinada hora (con una precisión de centésimas de segundo) debe comenzar la maniobra y apuntar la nave en una dirección en particular y encender el propulsor hasta aumentar la velocidad en esa dirección en una magnitud dada. El software controla la orientación de la nave, por medio del piloto automático, y con el motor principal modifica la velocidad hasta que lograr la velocidad final de la maniobra.

Si bien el sistema de navegación ejecuta las maniobras, es lo máximo que conoce del trayecto en sí, no conoce el objetivo final ni cuantas maniobras son necesarias para realizarlo. El software recibe instrucciones precisas del operador para la próxima maniobra (hora, dirección, cambio de velocidad, etc.) y llegado el momento la ejecuta, pero una vez terminada, para realizar otra maniobra deberá esperar a que el operador ingrese nuevos datos. Son los astronautas y el personal en Tierra los que tienen una visión general de la misión y van alimentando al sistema de navegación con parámetros de maniobras para lograr la trayectoria adecuada.

El software de la computadora de navegación podría encargarse de controlar todo el viaje, dirigir un trayecto largo sin requerir recibir instrucciones precisas de los astronautas periódicamente. Por ejemplo se le podría indicar que vaya hacia la Luna y entre en una órbita alrededor de ella y que el programa de la computadora realice los cálculos matemáticos para determinar que maniobras realizar y en qué momentos y la propia computadora los vaya ejecutando a medida que llegue el momento. Existen varias razones para optar por realizar los cálculos de forma externa al sistema de navegación y limitarlo a realizar una maniobra por vez.

Primero hay que tener en cuenta que la computadora seguía siendo algo novedoso y se prefería darle más control a los tripulantes y al personal en Tierra que depender de este equipo, y todas las acciones importantes de la computadora (y encender un motor lo era) debían ser aprobadas previamente por las tripulación. También existe el problema de que un programa que calcule la trayectoria más óptima, con la menor cantidad de maniobras posibles, ocuparía mucho espacio de memoria, tanto ROM como RAM, y son recursos que están muy limitados en este equipo. Aparte de lo que ocuparía el programa de existir hay que considerar el tiempo de desarrollo y testeo de un programa tan complejo sobre una computadora tan simple, más teniendo en cuenta que la gente encargada de realizar este tipos de cálculos los hacían sobre computadoras mucho más poderosas y hubiera sido muy complicado adaptarlos. En las computadoras de cálculo de misión se podía realizar muchos más cálculos por segundo que en esta pequeña computadora y podían aplicarse simulaciones de distintas opciones hasta obtener un resultado óptimo. El tiempo de ejecución de todo el análisis de la trayectoria sería muy largo y por la poca velocidad de este equipo sería necesario simplificar los cálculos y de esta forma se perdería precisión.

De todas formas, la computadora de las naves podía calcular algunas maniobras por si se perdía la comunicación con la Tierra, pero no lo hacía utilizando una simulación tan compleja como en los grandes centros de cómputo de la NASA que tenían en cuenta muchos factores, sino que realizaba aproximaciones en base a unos pocos datos. La computadora era capaz de obtener la maniobra para realizar la inyección transterrestre para poder abandonar la Luna y también ajustes de trayectoria durante el camino.

Recepción de parámetros para maniobras

Para que la computadora ejecute una maniobra es necesario que se le ingresen los parámetros, por ejemplo dirección, velocidad a ganar o tiempo estimado. Estos parámetros son calculados por computadoras en la Tierra y es necesario ingresárselos a la computadora de la nave para realizar la maniobra. Esto se podría realizar por medio del ingreso de estos valores directo en la computadora desde la interfaz remota en Tierra para cargar los valores sin intervención de los astronautas, pero un inconveniente de esto es que a veces la maniobra se debe ejecutar sin haber conexión de radio y el astronauta debería confiar que se hayan cargado los datos correctos en la computadora antes de que se perdiera la conexión. También es necesario pasarle a los astronautas distintas posibles maniobras todas

juntas para casos de emergencia así ellos pueden optar cuál realizar llegado el momento y recién ahí la ingresan en la computadora.

En un momento se estudió incluir una impresora en las naves a la que se le pudiera enviar textos por radio para proveer los parámetros por escrito a los tripulantes, pero se descartó para simplificar la misión. Lo que se terminó haciendo fue simplemente dictar los parámetros por radio y que un astronauta los anote en papel. Las hojas sobre las que escribe estos valores son formularios con los campos específicos para completar con esta información y así simplificar la escritura y la lectura. Para evitar errores, luego recibir la maniobra se verifica varias veces que los parámetros se hayan copiado correctamente, repitiéndole por radio los valores para que los confirmen remotamente. Luego de esta operación, bastante larga y tediosa, se tienen los parámetros que permiten realizar una maniobra en la computadora.

Maniobras especiales

La mayoría de las maniobras que deben realizar las naves son simples de ejecutar, apenas debe apuntar en una dirección y disparar alguno de los motores para ganar determinada velocidad.

El descenso del Módulo Lunar sobre la Luna es una maniobra larga y dividida en varias etapas y que requiere constante control por parte del software. La reentrada en la atmósfera del Módulo de Comando también es distinta al resto de las maniobras porque no se realiza en el vacío y la principal fuerza que se aplica sobre la nave es la de rozamiento con la atmósfera, la computadora debe rotar la nave para ir planeando, perdiendo velocidad de forma constante.

Otro caso especial son las maniobras que realiza el Módulo Lunar para volver a encontrarse con el Módulo de Comando en órbita luego de partir desde la Luna. Si bien lo realiza con maniobras orbitales similares a la mayoría, la diferencia es que los parámetros de estas maniobras no son calculados por las computadoras de la NASA y transmitidos por radio, sino que son resultados obtenidos por la propia computadora del Módulo Lunar. La diferencia con los cálculos de trayectorias obtenidos de las computadoras de la NASA, es que aquí la proyección se realiza de forma puramente geométrica, sin realizar avanzadas simulaciones con campos gravitatorios que afectan el movimiento de la nave. La computadora toma varias mediciones de la posición de la otra nave (obtenidos por el radar de aproximación al Módulo de Comando) y realiza una extrapolación, trazando una curva que se ajuste a las mediciones realizadas, y calcula cómo debería ser un impulso para convertir la curva que lleva el Módulo Lunar en una curva que se intersecte con la del Módulo de Comando en un determinado punto. Esta estimación es lo suficientemente buena como para que ambas naves se encuentren luego de orbitar la Luna a diferentes alturas y velocidades.

PLANIFICACIÓN DE LA MISIÓN

La planificación de una misión Apolo era muy estricta, no había mucho margen para improvisar, y el punto donde se produciría el alunizaje era lo que determinaba en gran medida cómo se realizaría toda la misión y era lo primero que se debía determinar para poder calcular el resto. Se debían tener en cuenta muchos detalles para seleccionar el destino, por ejemplo para poder tener comunicación por medio de ondas de radio de forma directa a las antenas terrestres, era necesario que el lugar de descenso estuviera sobre la cara visible de la Luna. Como la Luna da un giro completo sobre su eje en el mismo tiempo en que realiza una rotación alrededor de la Tierra, siempre nos muestra la misma cara, la otra permanece oculta. Además, el lugar de descenso debía estar ubicado lo más cercano posible del ecuador lunar porque por allí es por donde pasa el plano en el que la Luna orbita a la Tierra, mientras más se alejen del ecuador lunar más habrá que modificar el plano orbital, y es una maniobra que requiere mucho combustible.

De las posibles zonas lunares los científicos seleccionaban los puntos en que pudiera haber minerales interesantes para estudiar pero que a la vez parecieran zonas regulares para que la nave se pudiera posar sobre el suelo. De todas formas, con la información con la que se contaba en aquella época, no era posible conocer mucho sobre la superficie, no había certeza de que una zona no estuviera llena de rocas que hicieran imposible el contacto con la superficie, ni siquiera se conocía con exactitud la altitud en cada punto. La composición del suelo y su densidad tampoco era conocida.

Una vez que se seleccionaba el destino la planificación de la misión requería de muchos cálculos para obtener las trayectorias óptimas para realizar el viaje. Los cálculos incluyen los campos gravitatorios de la Tierra y la Luna que afectan a la nave en cada punto de su trayectoria, y así se debería evaluar en qué punto de una órbita terrestre lanzar la nave y con qué velocidad para alcanzar la Luna con el menor gasto de combustible. Pero también hay que recordar que no es un sistema estático: la Luna se mueve alrededor de la Tierra y, si se quiere enviar una nave a la Luna, se debe apuntar no hacia donde se encuentra ahora sino hacia donde estará la Luna en el tiempo que le toma a la nave llegar hasta ella, pero para saber cuánto tiempo le llevaría a la nave (y entonces hacia qué posición de la Luna apuntar) se necesita saber la trayectoria que seguirá. No existe una fórmula matemática para resolver esto de forma exacta, sino que se recurre a simulaciones por computadora de este sistema, que realizan millones de operaciones matemáticas hasta obtener una aproximación lo suficientemente útil y se evalúan distintas trayectorias hasta obtener la más óptima.

En la planificación se tenía en cuenta el peso de todo lo que se llevaba, lo que se abandonaba en el camino, en qué etapa se abandona y el peso de las muestras de suelo a traer. En los cálculos se planificaba esto con gran detalle para poder estimar cuanto combustible cargar en el cohete y en cada nave. Mucho antes de partir ya se habían planificado todas las maniobras a realizar y durante la misión no era posible apartarse mucho de estos cálculos.

Para este fin se utilizaban los mainframes más grandes de la época, durante unos años los IBMs 7094-11 y luego IBMs System/360 modelo 75 [10]. Al comienzo del programa espacial se utilizaban estas computadoras para realizaban estimaciones de las capacidades del cohete y las naves, y en base a eso se comenzó con el diseño y fabricación de cada componente. Luego se continuaron usando para calcular cómo realizar una misión real.

A pesar de todos los cálculos realizados para una misión y de la gran planificación específica para cada lanzamiento, el firmware de la computadora de las naves era independiente de todo esto. Estos programas deberían poder usarse en cualquiera de las misiones típicas del programa Apolo y todas las particularidades de la misión en curso, por ejemplo la cantidad de combustible con la que parte la nave, se le debían poder cargar en variables en RAM antes del lanzamiento y las maniobras se le irían cargando a medida que se realizaban.

CONCLUSIONES

Lo que se vio hasta aquí es una descripción del sistema de navegación en general, tanto el software como el hardware. En las secciones anteriores se vio porque era necesario llevar una computadora, o dos en este caso, para realizar el viaje ida y vuelta hacia la Luna, repasando las características de un vuelo espacial y también se dio una descripción general del sistema, la tecnología utilizada y decisiones de diseño.

En la próxima parte nos centraremos en cómo se utilizaba la computadora en una misión para guiar a las naves hacia la Luna y de la Luna de vuelta a la Tierra.

BIBLIOGRAFÍA

[1] M. S. Johnson (1971). *MIT's role in project Apollo, Volume V: The software effort*, Charles Stark Draper Laboratory, Marzo 1971.
[2] D. A. Mindell (2008). *Digital Apollo*, MIT Press
[3] J. E. Tomayko (1998). *Computers in Spaceflight: The NASA Experience.* Wichita State University. http://www.hq.nasa.gov/office/pao/History/computers/contents.html (11 de junio de 2016).
[4] E. C. Hall (1972). *Apollo guidance, navigation and control Volume III: Computer subsystem*, Charles Stark Draper Laboratory, Agosto 1972.
[5] C. A. Muntz (1965). *User's guide to the Block II AGC/LGC interpreter*, MIT instrumentation laboratory.
[6] F. O'Brien (2010). *The Apollo Guidance computer Architecture and operation*, Springer Praxis.
[7] D. Eyles (2004). *Tales from the lunar module guidance computer*: http://www.doneyles.com/LM/Tales.html (20 de mayo de 2016).
[8] D. G. Hoag (1976). *The history of Apollo on-board guidance, navigation, and control*, Charles Stark Draper Laboratory.
[9] W. D. Woods (2011). *How Apollo Flew to the Moon*, Springer Praxis.
[10] IBM (2016). *Real-Time Computer Complex*. IBM Archives. http://www-03.ibm.com/ibm/history/exhibits/space/space_realtime.html (1 de junio de 2016).

LA COMPUTADORA DE LAS MISIONES APOLO
PARTE IV: LA COMPUTADORA EN LA MISION

Marcos Leguizamón†

En esta sección se describe como era la utilización de la computadora dentro de una misión espacial para la que fue diseñada. Durante el viaje de ida y vuelta a la Luna, que duraba en total una semana para una misión básica, los astronautas debían interactuar con la computadora para realizar las distintas maniobras y para poder evaluar el estado de la trayectoria. Toda la información que se ingresaba y se leía de la computadora consistía en números que sólo tenían sentido si se estaba familiarizado con el programa en ejecución, que variaba dependiendo de la etapa del viaje y de la maniobra que se estaba realizando. Por esa razón los astronautas debían estar muy interiorizados en el funcionamiento de este sistema para poder realizar el viaje completo.

PREPARACIÓN DE LA MISIÓN

La misión comienza con el despegue del cohete desde una plataforma de lanzamiento en Cabo Cañaveral, pero la preparación comienza meses antes en un gigantesco hangar a 5 km de distancia, donde se toman las distintas naves y etapas del cohete que vienen desde todos los rincones de los Estados Unidos, algunas incluso cruzando el Canal de Panamá, y se ensamblan formando una estructura de 110 metros de alto que contiene, de abajo hacia arriba, la primera, segunda y tercera etapa del cohete Saturno V, luego el Módulo Lunar seguido del Módulo de Servicio que está unido al Módulo de Comando ubicado en el extremo superior de todo el grupo. Junto al cohete se ensambla la torre de lanzamiento, de 136 metros de alto, que sostiene al cohete hasta el momento del despegue. Cuando está todo completamente ensamblado pero sin combustible (que representa la mayor parte del peso), se traslada en un gigantesco vehículo con orugas a la plataforma de despegue [1].

Mientras permanecen en la plataforma, tanto el cohete como el Módulo de Comando están conectados a la torre de despegue por varios conjuntos de cables, y desde la

† RTI Latina S.R.L., leguizamon@gmail.com

central de control se monitorea el estado de los numerosos sistemas eléctricos de las naves. Por otro lado, el Módulo Lunar permanece desconectado e inaccesible, protegido en un compartimiento especial, y recién durante el viaje a la Luna se activará porque su ambiente ideal es el vacío.

Semanas antes del despegue se realiza un ensayo general con la tripulación en sus puestos en el Módulo de Comando con los trajes espaciales y se verifica el correcto funcionamiento de los sistemas (alimentación eléctrica, control de atmósfera, comunicación, etc.) y se pone a prueba la computadora. Se corre un programa que verifica el correcto estado del hardware y lee el contenido completo de la ROM calculando su checksum, que es mostrado en el visor para que la tripulación lo compare con el que le corresponde a la versión actual del firmware. El programa también escribe y lee todas las palabras de la memoria RAM de núcleos magnéticos para comprobar que no haya bits fallados. La computadora del Módulo Lunar, ahora inaccesible, fue verificada antes de montar la nave dentro del Saturno y recién en el espacio se podrá volver a probar.

Ventana de lanzamiento

En una misión espacial se conoce como ventana de lanzamiento el plazo en que debe despegar el cohete. Si por alguna razón, generalmente de índole climática, no fuera posible realizar el lanzamiento dentro de ese plazo, sería necesario esperar a la siguiente ventana. Esto es así porque las misiones espaciales están diseñadas para aprovechar distintas configuraciones de los cuerpos del sistema solar, y las ubicaciones de estos varían continuamente.

En el caso de una misión Apolo, los requerimientos que se deben cumplir tienen que ver con la iluminación en la superficie lunar, la distancia de la Luna con respecto al plano ecuatorial de la Tierra y la ubicación de distintas fases importantes de la misión en relación a las antenas terrestres. La distancia de la Luna a la Tierra no es importante porque no varía mucho durante su órbita, distinto es el caso de una misión a Marte porque la diferencia entre la mayor y la menor distancia es enorme.

La iluminación en el lugar donde la nave descenderá y los astronautas explorarán es importante, porque de noche es casi imposible ver nada y la temperatura es muy baja, y cuando el Sol está muy alto en el cielo la luz es excesiva y lo mismo la temperatura. El horario ideal para alunizar y explorar es a la mañana lunar, donde al ser iluminados de forma lateral se producen sombras muy útiles para identificar características del paisaje. Como un día Lunar (el tiempo en que el Sol da una vuelta completa visto desde un punto en la Luna) es de aproximadamente 29 días terrestres, las condiciones requeridas de iluminación duran varios de nuestros días pero pasado esas *horas lunares* hay que esperar alrededor de un mes para tener las mismas condiciones.

Cuenta regresiva

El día del lanzamiento ya está todo listo para despegar: unos días antes se llenaron los distintos tanques de combustible y oxidante del Saturno V utilizando decenas de camiones para esto. Faltando 28 horas para el lanzamiento comienza la cuenta regresiva, que al llegar a 0 significará el ascenso del cohete. Cada instante de la cuenta regresiva se conoce como T-n (por ejemplo T menos 15 minutos o T menos 9 segundos) y a cada momento de esta cuenta le corresponde una acción o la verificación de algún componente a cargo de los muchos ingenieros encargados de cada sistema del cohete. Si por alguna razón hay un retraso, se puede detener la cuenta regresiva y continuar luego, pero sólo cuando todavía se está lejos del 0, en los instantes previos al lanzamiento se inician procesos críticos que no pueden ser retrasados.

En T-2 horas se ubica la tripulación en el Módulo de Comando y se realizan los chequeos finales de todos los sistemas. Unas horas antes la tripulación suplente puso cada uno de los cientos de interruptores, llaves, válvulas y fusibles de la nave en la posición adecuada y verificó el correcto funcionamiento de la computadora. Los tres astronautas se ubican en los asientos, uno al lado del otro, con el piloto del Módulo de Comando en el centro, donde se encuentran los principales controles de navegación y la interfaz de la computadora. El comandante se sienta en la silla izquierda, donde está la palanca de mando que controla el desplazamiento de la nave. El piloto del Módulo Lunar no permanece ocioso, sino que queda frente a muchos interruptores e indicadores para supervisar, como los controles de cantidad de combustible, alimentación eléctrica y comunicación por radio, entre tantos otros. Ahora se cargan las coordenadas del cohete en Tierra y se ejecuta un programa que hace que la unidad inercial, en lugar de permanecer apuntando en la misma dirección en el espacio a medida que la Tierra gira sobre su eje, se mantenga alineada siempre con la gravedad terrestre y así tiene un sistema de referencia que pasa siempre por el centro de la Tierra. Sólo el comandante puede ver el exterior por una pequeña ventana porque el resto están obstruidas por la torre de escape que se encuentra sobre el Módulo de Comando y se descartará recién al abandonar la Tierra.

En T-60 segundos se corta la alimentación eléctrica desde la torre de lanzamiento y se pasa al suministro eléctrico interno de la nave.

En T–17 la unidad inercial del sistema de guiado del Saturno V se libera, se deja de ajustar su orientación según la gravedad terrestre, a partir de este momento se tomará la orientación actual y se anuncia que “el guiado es interno”.

Al llegar a T–9 se inicia la secuencia de ignición de los motores de la primera etapa del Saturno V, de a uno por vez, consumiendo cada uno hasta 3 toneladas de kerosene por segundo, pero no comienza a ascender porque el cohete está fuertemente sujeto por varios soportes desde abajo. En este momento el comandante carga el programa para esperar el despegue y pone la mano cerca a la palanca para abortar el despegue, que controla el mecanismo de eyección de los astronautas en caso de que no se activase de forma automática. Cuando la cuenta regresiva llega a 0 se pasa a la etapa de ascenso.

Etapa de ascenso

El cohete comienza a elevarse cuando los soportes que mantienen al cohete sujeto a la base se sueltan y en este mismo momento se liberan los brazos laterales que unen al cohete y al Módulo de Comando con la torre de lanzamiento y se desconectan todos los cables: ahora el cohete es completamente autónomo y la única comunicación entre los sistemas del cohete y el control el Tierra es por radio. En ese mismo momento, y justo antes de cortar los cables que unen la nave a la torre, desde esta se le envía una señal a la computadora que hace que la ejecución pase al programa que se encarga de monitorear el ascenso del cohete de forma pasiva. En caso de que por algún problema la computadora no recibiera esta señal de despegue, el comandante presiona la tecla ENTER porque la computadora se encuentra en un estado que solamente necesita ese evento para pasar a este programa. Al iniciarse el programa de monitoreo de ascenso, la computadora comienza a contar el tiempo a partir de ese instante y se toma este reloj (con el tiempo transcurrido desde el lanzamiento) como la hora de referencia para usar durante toda la misión, en lugar de utilizar la hora local de algún punto en la Tierra.

El Saturno V muy lentamente va ganando velocidad y le lleva 10 segundos para pasar la altura de la torre de lanzamiento y para este momento ya consumió en combustible un 4 % de todo su peso. Una vez superada esa altura, el control de la misión pasa a la central en Houston, Texas (a 1.500 km de Cabo Cañaveral), donde los controladores de vuelo supervisaran toda la misión hasta su regreso a la Tierra. Si bien el cohete despega en dirección vertical, a medida que asciende lentamente va inclinándose hacia el este, porque su meta no sólo es llegar hasta 190 km de altura, sino hacerlo a una velocidad horizontal de 8 km/s para poder entrar en órbita, por eso al llegar a esa altura ya debe viajar de forma paralela al suelo (Figura 1).

A medida que el cohete consume combustible va perdiendo peso y, como también a mayor altitud menor es la presión atmosférica (y la resistencia del aire), la fuerza de los motores producen una aceleración cada vez mayor. Luego de ascender durante un minuto, estando a 8 km de altura ya alcanza una velocidad de Mach 1 y unos segundos después se alcanza la máxima presión dinámica (conocida como Max Q), el momento en que el cohete debe soportar la mayor fuerza de resistencia a causa de su movimiento en el aire, y pasado este punto la presión desciende. La primera etapa empuja en total durante unos 2 minutos y medio hasta agotar su combustible al llegar a aproximadamente 70 km de altura a una velocidad de 8500 km/h con una aceleración de 4 G, y en esta etapa se desprende del resto del cohete. En este momento también se eyecta la torre de escape que está sobre el Módulo de Comando y recién ahora la tripulación puede ver a través de las cinco ventanas.

La segunda etapa se enciende para continuar el ascenso y permanece empujando durante 6 minutos y medio a un cohete mucho más liviano y con una presión atmosférica muy tenue, pero nunca llega a acelerar más de 2 G porque tiene una potencia mucho menor a la de la etapa ya descartada. Al acabarse el combustible de esta fase a 175 Km de altura ya casi se alcanzó la altura final y el movimiento es prácticamente en paralelo al suelo aunque

la velocidad no es la necesaria para entrar en órbita, por eso se necesita todavía el empuje la tercera etapa, luego de desprenderse de la segunda.

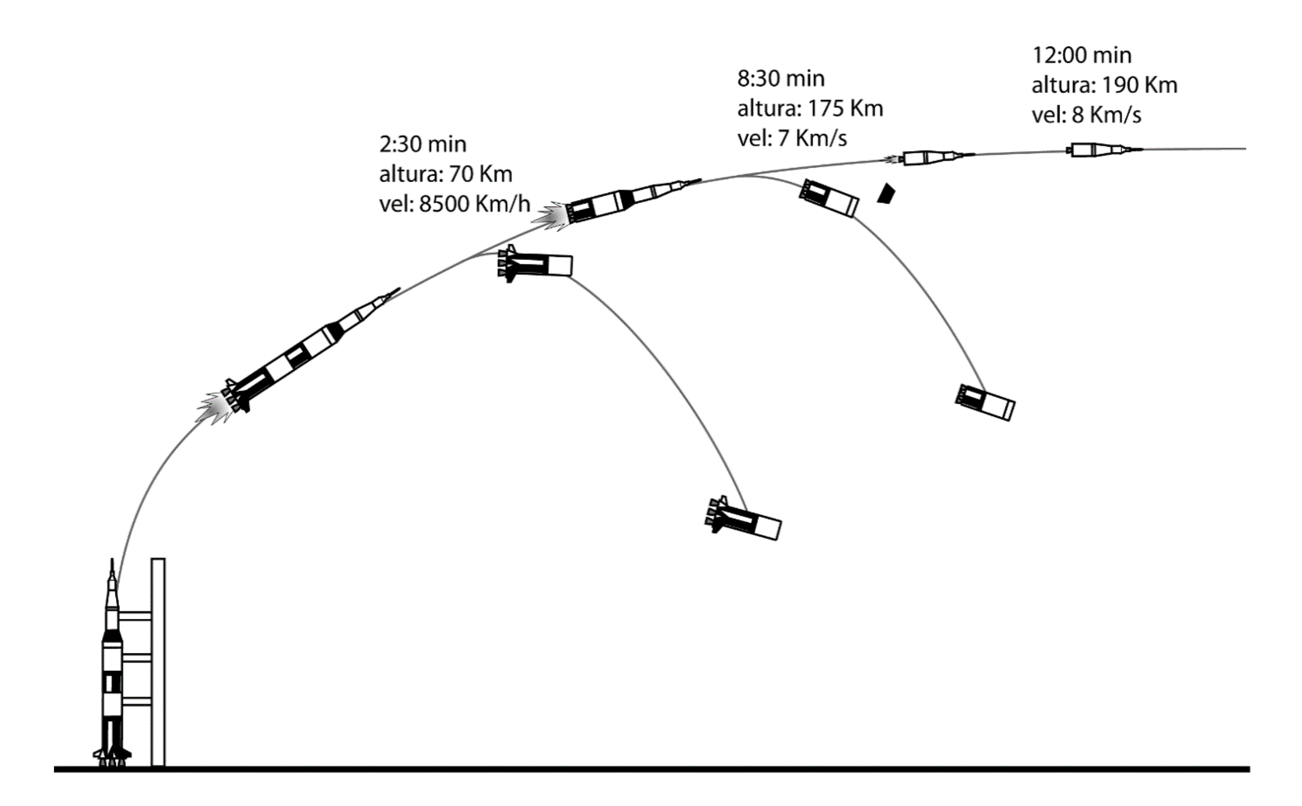


Figura 1. Entrada en órbita del Saturno V. Luego de deshacerse de las dos primeras etapas queda orbitando la Tierra a 190 km de altura.

Al darse ignición a la tercera etapa se sigue siempre en dirección casi horizontal acelerando durante 2 minutos de forma tenue (0,5 G a 0,75 G) hasta alcanzar la velocidad de 8 km/s. En ese momento el cohete entra en órbita y el motor se apaga pero no se descarta esta etapa final del Saturno V porque todavía queda la mayor parte de su combustible y será necesario utilizarlo luego para ir hacia la Luna.

Aproximadamente luego de 12 minutos de haber despegado, el cohete, que ahora consta de sólo la tercera etapa más las dos naves en su extremo, está en órbita alrededor de la Tierra a 190 km de altura dando una vuelta completa al planeta en 90 minutos.

Durante todo el despegue la tripulación no realiza ninguna tarea, solamente están expectantes de que se active el mecanismo de eyección automático en caso de detectarse una anomalía. En la computadora corre el programa que monitorea el ascenso mostrando en los tres visores la velocidad, velocidad de ascenso y altitud para que la tripulación compare estos valores con una tabla que tienen en papel con los valores que deberían ir alcanzando en la trayectoria normal, así si detectan que ocurre un problema deberían abortar la misión manualmente. En caso de una emergencia, para escapar del Saturno V mientras se asciende con la primera etapa, se utiliza el cohete de eyección que está en la punta de la nave y aleja al Módulo de Comando que luego abre sus paracaídas y pone a salvo a la tripulación. Si hubiera que abortar la misión durante las otras etapas se entra en un modo que hace que el

Módulo de Servicio se separe del Saturno y se aleje para luego hacer una reentrada a la atmósfera.

La unidad inercial del Módulo de Comando deja de rotar junto con la Tierra al momento del despegue y quedará apuntando en la dirección en que estaba la nave en Cabo Cañaveral en el instante del despegue. Se usará este sistema de referencia (con el origen en el centro de la Tierra) durante una parte de la misión, luego se cambiará a otros según la etapa.

COMPUTADORA DE CONTROL DEL SATURNO V

El Saturno V cuenta con su propia computadora de control, conocida como LVDC (Launch Vehicle Digital Computer), notablemente más grande que la computadora de las naves Apolo y con una arquitectura muy diferente [2, 3, 4]. Esta computadora fue creada por IBM y desciende de la usada para controlar al misil Titan. Su velocidad es muy lenta porque, para simplificar el hardware, procesa las instrucciones de forma serial, un bit por vez, y por eso apenas realiza unas 7000 operaciones por segundo. También se usó una versión de esta computadora en las cápsulas espaciales del proyecto Gemini, el programa espacial anterior al Apolo, donde se pusieron a prueba muchas de las técnicas para poder ir a la Luna. Durante el transcurso del proyecto Gemini, entre 1965 y 1966, el MIT estaba trabajando en el desarrollo de la computadora para el programa Apolo, pero para algunas misiones se necesitaba contar con una computadora para controlar las naves, siempre en la órbita terrestre, aunque con maniobras más complejas que las usadas hasta ese momento, sobre todo encuentros entre dos naves, técnica que era necesaria poder dominar bien antes de pasar al programa Apolo.

Esta fue la primera computadora de IBM completamente hecha con semiconductores, aunque no utilizando circuitos integrados sino transistores individuales. Estaba distribuida en un anillo de 6,6 metros de diámetro y 1 metro de alto en la parte superior de la tercera etapa del Saturno V por lo que era parte de la estructura del cohete y era visible desde afuera al representar casi el 1 % de su altura. Trabajaba con palabras de 26 bits pero las instrucciones eran de apenas 13, dos por cada palabra. En la memoria se guardaban 28 bits por palabra porque tenía 2 bits de paridad y utilizaba una memoria de núcleos magnéticos con 16 mil palabras, pero no había ninguna memoria ROM sino que toda la información, ya sea el programa como los datos, estaba dentro de la memoria de núcleos magnéticos, por lo cual el programa podía modificarse a sí mismo, ya sea voluntariamente o por error. Como en las memorias de núcleos magnéticos se suele usar lectura destructiva, lo que hace que mientras se lee cada instrucción del programa en ejecución se pierda momentáneamente el contenido de esa palabra de la memoria hasta que se lo rescribe, podría corromperse una palabra si se apagara la computadora en el momento equivocado. Para evitar ese caso, en IBM inventaron una forma más compleja de manejar la memoria que permitía lectura no destructiva, se leía la información de cada núcleo sin afectar su contenido.

En el cohete hay tres computadoras iguales controlando el vuelo. Las tres corren el mismo programa y las salidas generadas por los tres equipos pasan por un circuito que compara las tres órdenes obtenidas y genera una sola salida que es la salida seleccionada por la mayoría. En circunstancias normales las tres computadoras deberían generar las mismas órdenes en el mismo momento, si el circuito de validación detectara que una de las computadoras da una orden distinta a las otras dos, ignoraría esta computadora y la desactivaría, porque significaría que algo malo le ocurrió y el vuelo está en peligro si se siguen órdenes de esta computadora. Esta arquitectura, llamada TMR (Triple Modular Redundancy) se utiliza en los ambientes de control crítico aeroespacial o militar y en la actualidad se fabrican chips que en su interior tienen la lógica repetida tres veces para este fin.

Esta computadora ejecuta un programa que mantiene la trayectoria pre programada con su sistema de navegación inercial, realizando ajustes 25 veces por segundo. Si bien su sistema de navegación inercial durante todo el trayecto conoce la orientación y aceleración en cada eje y de estos deduce velocidad y posición, se optó por hacer que los primeros tres minutos del vuelo se realizaran de forma totalmente pre programada sin realizar ajustes según lo medido por los instrumentos, en lo que sería un sistema de lazo abierto sin correcciones. En este caso la potencia y dirección que se da a cada motor seguiría los tiempos pre programados sin importar variables como la velocidad del viento o desviaciones por pequeños errores, y de esta forma se aseguraban que la computadora no tomaría ninguna decisión peligrosa estando el cohete cerca del suelo. Recién pasado este tiempo el sistema de navegación comienza a realizar los ajustes para que la trayectoria se aproxime lo más posible a la trayectoria ideal. De todas formas, en todos los lanzamientos el personal en Tierra monitorea con telescopios la trayectoria del cohete y si el cohete pierde el control le envía por radio el comando de autodestrucción (el cohete cuenta con explosivos para este fin).

Una diferencia importante de esta computadora a las usadas en las naves, es que no es operada por humanos, no tiene visores ni teclado, sino que controla al cohete de forma autónoma. Recibe instrucciones por radio desde Tierra con los últimos parámetros para la trayectoria que debe realizar, pero existen básicamente tres maniobras que debe controlar, el ascenso inicial, la estadía en órbita alrededor de la Tierra y luego la inyección translunar para dirigirse a la Luna. Las naves, en cambio, deben realizar maniobras más variadas y complejas que requieren interacción con los tripulantes.

Motores del Saturno V y de las naves

El cohete Saturno tiene en total 11 motores en sus tres etapas, en el Módulo de Servicio hay un solo motor principal y en el Módulo Lunar existen dos motores, uno usado para descender sobre la Luna y que es abandonado allí y otro para despegar desde la Luna. Los motores del Saturno son notablemente distintos a los de las naves y se controlan de forma muy diferente, por ejemplo en las dos primeras etapas del Saturno se usan de a 5 al

mismo tiempo. Cada motor del Saturno tiene una complejidad importante, y su sistema de control automático debe controlar la ignición, regular con válvulas y bombas la presión de combustible y de oxidante que entran en la cámara de combustión, monitorear la temperatura y controlar que todos los parámetros estén en los rangos permitidos. En la primera etapa se usaba de combustible kerosene y en las otras dos hidrógeno líquido, pero en todas había un tanque de oxígeno líquido. El sistema de control debe administrar el flujo de combustible y oxígeno para evitar que se consuma más de uno que del otro. Si al consumirse todo el combustible quedara en el tanque de oxígeno un peso importante de este gas, o lo mismo si lo que se consumió primero fue el oxígeno y quedó mucho combustible sin poder usarse, significaría que se cargó durante todo el trayecto ya realizado un peso importante e innecesario y difícilmente se pueda cumplir su objetivo de esta forma tan ineficiente. Por eso es importante que el sistema de guiado administre las cantidades de los tanques para realizar todo el empuje de forma óptima y para hacer todo esto se cuenta con distintos tipos de sensores en los tanques y los motores.

Los motores de las naves no necesitan ser tan eficientes porque no deben empujar un peso tan grande como el Saturno V y por eso se le dio más importancia a la simplicidad en su diseño. Como combustible se usaban los químicos hidracina y tetróxido de dinitrógeno. Estos líquidos son conocidos como hipergólicos y tienen la propiedad de que al entrar en contacto entre sí automáticamente hacen ignición, por eso el motor se diseña para que abriendo una válvula se combinen estos dos químicos en la cámara de combustión y comiencen a producir empuje, sin necesidad de mecanismos para ignición, y ni siquiera hacían falta bombas porque se usaba un tanque con helio a presión para empujar los químicos por la tobera. La computadora podía controlar un motor con tan sólo un bit, simplemente abriendo o cerrando el paso de los líquidos, y luego era cuestión de regular el tiempo de encendido y la orientación de la tobera. Era muy importante que estos motores no fallaran porque si no la nave no podría desplazarse y esto en la mayoría de los casos significaba que la tripulación no podría volver a la Tierra. La simplicidad de estos motores hacía que los tripulantes los pudieran controlar manualmente en caso de fallas del sistema automático, tan solo manipulando la válvula adecuada.

La contra de este tipo de combustible hipergólico era que, además de ser altamente tóxico y presentar graves riesgos en su utilización por el personal en Tierra, su combustión era muy corrosiva con los metales del motor, esto hacía que una vez que se encendía el motor comenzara a deteriorarse, por lo cual no era posible probar los motores en Tierra sino que se encendían por primera vez estando en el espacio.

Control manual del Saturno V

El cohete Saturno V debe realizar sus maniobras utilizando su propio sistema de navegación, pero aun así existe una forma para controlar su ascenso desde la computadora del Módulo de Comando en caso de que su sistema de navegación no funcionara correctamente [5]. A pesar de contar con mucha redundancia en su electrónica, la

plataforma inercial del Saturno V era una sola y si se descomponía la única forma de guiarlo sería desde el Módulo de Comando. En este caso el sistema de guiado del cohete detecta que su plataforma inercial tiene un problema y envía una señal que enciende una luz de alerta en el panel de la nave. Cuando el comandante ve esta luz, activa un interruptor que hace que la información de la plataforma inercial de la nave vaya hacia el sistema de guiado del cohete para que este la tome como referencia. Si el propio cohete no detecta un problema no hay forma de iniciar esta secuencia manualmente.

La computadora del Módulo de Comando no tenía un programa para dirigir al Saturno, por eso es el comandante el que debe guiar al cohete usando la palanca de dirección, a través de la computadora, para indicar la dirección de ascenso comparando lo que ve en el visor de la computadora con una tabla con la correcta trayectoria a seguir. Nunca se llegó a poner en práctica nada de esto en ninguna misión pero igual cada tripulación practicó este control del cohete para evitar tener que abandonar todo el viaje por una pequeña falla en el Saturno V.

Orbita de estacionamiento

Una vez que terminó la etapa de ascenso, en lugar de ir directo hacia la Luna, las naves y lo que resta del Saturno V quedan en órbita alrededor de la Tierra. Permanecerán así hasta verificar que está todo en orden para emprender el viaje a la Luna, el cuál que debe partir de un punto en particular de la órbita [1]. Desde que se entró en órbita los acelerómetros ya no medirán ninguna aceleración durante la misión con la excepción del tiempo que permanezcan en la superficie lunar y durante los breves momentos en que los motores empujan a las naves, que son apenas minutos en comparación con los días que dura la misión. El principal mecanismo de la computadora para calcular su aceleración, velocidad y posición en cada momento es usando su modelo matemático de los campos gravitatorios que afectan a la nave.

El control de la misión por el momento permanece a cargo de la computadora del Saturno, a su vez controlada desde Tierra, que, por ejemplo, mantiene la orientación del cohete en la órbita. Los astronautas vuelven a verifican el correcto estado de todos los sistemas luego de haber soportado el ascenso a órbita y comprueban el funcionamiento de la nave en el vacío. Se desprenden las coberturas externas del sextante que lo protegían en la atmósfera, ahora ya se puede observar el exterior con él y se utiliza para orientar y calibrar la unidad inercial de la nave.

Durante la órbita, como se está a corta de distancia de la Tierra, solamente se tiene conexión por radio con Houston durante unos pocos minutos, cuando están cerca de alguna antena repetidora, por eso se aprovechan esos instantes de conexión por radio para hacer los ajustes que necesiten supervisión.

Comunicación por Radio

Para poder llevar adelante la misión no alcanza con los astronautas y todo lo que llevan consigo, sino que requieren ayuda de los controladores de vuelo en Houston y para eso es necesario utilizar la radio. La comunicación por radio entre la nave y una antena en la Tierra se puede establecer cuando no hay ningún obstáculo en el medio y el principal obstáculo es la propia Tierra que oculta la mayor parte del cielo a una antena ubicada en su superficie. Durante el tiempo que la nave está orbitando a baja altura, cada antena en la Tierra sólo puede comunicarse con la nave un breve tiempo antes de que pierda la conexión cuando la nave se oculte detrás del horizonte, y sería necesario disponer de una gran cantidad de antenas para tener comunicación permanente. Una vez que la nave ya se alejó lo suficiente de la Tierra, a distancias comparables al diámetro del planeta, la señal de radio que llega es más débil pero se puede comenzar a utilizar unas enormes antenas parabólicas direccionables de 30 metros de diámetro que apuntan con precisión a la posición de la nave, lo que permite captar mejor la señal y, además, cada antena puede abarcar gran parte del cielo gracias a su capacidad de orientarse. Usando tres de estas antenas alrededor del planeta, en Estados Unidos, España y Australia, con aproximadamente 120 grados de diferencia en la ubicación de cada una, se puede abarcar todo el cielo. A medida que la nave se mueve y la Tierra rota sobre su eje, se va seleccionando una de las tres antenas y de esta forma se pasa a tener comunicación de forma permanente entre las naves y Houston. Esta red de antenas interconectadas fue creada por la NASA en los 60 con el nombre de Deep Space Network (red de espacio profundo) y además usarla para el programa Apolo sirve (y se sigue utilizando) para establecer comunicación con sondas interplanetarias, como las Voyager o el robot Curiosity [6].

Las naves disponen de dos sistemas de radio con distinto tipo de antena, las omnidireccionales que están fijas y sólo requieren estar ubicadas en el lado de la nave que apunta a la Tierra y después está la antena parabólica de alta ganancia, que se orienta automáticamente hacia la antena en Tierra, sin usar la computadora sino que por medio de un circuito que la gira hasta obtener la mejor señal. Esta antena parabólica dispone de un mayor ancho de banda y puede transmitir mayor cantidad de información. La antena omnidireccional, en cambio, está limitada en lo que puede transmitir y está para usarse en caso de que falle el sistema de orientación de la antena parabólica mientras se soluciona el problema.

Existe comunicación por voz en ambas direcciones entre las naves y Houston, también desde el espacio se puede enviar señal de video en los pocos momentos en que se enciende la cámara de televisión y, además, se envía continuamente información digital de telemetría de las naves hacia la Tierra.

Gracias a la comunicación por radio, en Houston disponen de más información de sensores en la nave que la que poseen los propios tripulantes, porque no hay espacio para todos los indicadores en ninguna de las naves, por ejemplo en Tierra monitorean el latido del corazón de cada astronauta y de la computadora obtienen información interna en donde

pueden ver más rápido el estado de algún periférico de lo que llevaría configurar la computadora en la nave para poder ver ese estado en el visor. De los valores transmitidos desde la computadora saben siempre la posición, velocidad y aceleración que calculó el sistema de navegación y la pueden comparar con la observada desde Tierra.

También se puede enviar información digital en sentido inverso, desde Houston a la nave, pero esto es un acceso exclusivo a la computadora, no se puede controlar nada de la nave por radio, e incluso los tripulantes pueden deshabilitar con un interruptor la entrada de ratos por radio hacía la computadora. En Houston sólo se obtiene información desde la nave, no se controla el vuelo ni nada de las naves, están limitados a enviarle indicaciones por voz a la tripulación. La excepción es el control del cohete Saturno, que se le envía periódicamente instrucciones por radio a la computadora y las actualizaciones de las coordenadas que se enviaban a las computadoras de las naves. En las misiones no tripuladas, aprovechando que no tienen el peso de los tripulantes y los asientos, se instalan controles extra dentro de las naves para poder operarlas de forma remota, pero en las tripuladas todo debe pasar por los astronautas.

Al disponer de una buena comunicación durante la mayor parte del viaje y al ser capaces de monitorear el estado de todos los sistemas de la nave y el funcionamiento de la computadora, con el tiempo se abandonó la práctica de que los astronautas se turnaran para dormir y que siempre hubiera al menos un astronauta despierto para controlar el correcto funcionamiento de la nave. Lo que resultó más práctico fue que los tres durmieran a la vez y si desde Tierra se detectaba alguna anomalía se los despertaba por radio.

CENTRO DE CONTROL DE MISIÓN EN HOUSTON

En Houston toda la misión es controlada por el Director de Vuelo que coordina un grupo grande de expertos en las distintas especialidades necesarias durante el vuelo [7, 8]. La sala principal donde está el director es la Sala de Control de Misión y cada puesto aquí tiene un nombre, por ejemplo BOOSTER es el que controla y verifica todo lo relacionado con el Saturno V, FIDO es el que supervisa la trayectoria de la nave, toma las últimas mediciones de la posición y velocidad de la nave junto con el peso estimado por el consumo de combustible y calcula con las computadoras del centro de cómputo en tiempo real como se debe realizar cada maniobra. GUIDO es el encargado en Tierra del sistema de navegación de la nave, supervisa su correcto funcionamiento y lo compara con las mediciones hechas desde Tierra. Solamente una persona se comunica con los astronautas en el espacio, este puesto se llama CAPCOM y es ocupado por otro astronauta que es el encargado de transmitir por radio las instrucciones que decide el Director de Vuelo en base al análisis del resto de los especialistas de cada área.

Toda la información que se recibe y transmite por radio por los canales de datos primero pasa por una computadora dedicada exclusivamente a la comunicación: una UNIVAC 494. Esta computadora verifica el correcto formato de los paquetes de datos recibidos y luego la información digital obtenida pasa a la computadora principal, la que

controla toda la misión en Houston y en este caso es una IBM System/360 con 1 MB de RAM que está ubicada en el centro de cómputo de tiempo real (RTCC). La IBM verifica que la información obtenida este dentro de los rangos esperables, genera las alarmas necesarias en casos de errores y continuamente calcula la posición de las naves en base a la información más actualizada utilizando la simulación de los campos gravitatorios y la física propia de la nave. Se utilizan dos computadoras IBM a la vez continuamente haciendo exactamente lo mismo, aunque solamente una genera las salidas de información. Si una de las dos computadoras se descompone se puede apagar y pasar todo el control a la otra, la redundante, porque en caso de no tener ninguna computadora corriendo no es posible continuar con la misión con normalidad. Además de estas dos computadoras hay tres más iguales disponibles en los momentos en que fuera necesario evaluar maniobras nuevas, así se simula la nueva maniobra en una de estas computadoras y luego se decide si se usan estos resultados.

En la Sala de Control de Misión los controladores de vuelo se sientan delante de lo que podrían parecer terminales de computadora. En realidad son consolas, fabricadas por Philco, con una pantalla de video blanco y negro, auriculares con micrófonos y múltiples interruptores con luces y, en algunos casos, un marcador de teléfono a disco. Utilizando los interruptores se puede seleccionar alguno de los muchos canales de comunicación de voz entre distintos puestos dentro de la sala, con especialistas de afuera o hasta con los astronautas. También se puede elegir entre decenas de señales de video para mostrar en la pantalla, por ejemplo la cámara que transmite el despegue o las de las naves (cuando están encendidas), pero también se pueden ver pantallas con información obtenida desde la telemetría de las naves o del Saturno. Estas pantallas con datos se transmiten desde una sala especial dedicada a esta tarea, donde unos dispositivos, similares a lo que luego serían las placas de video, generan una señal de video con información digital obtenida de la computadora IBM principal que controla la misión. A diferencia de las computadoras personales que aparecerían en la siguiente década (como la Apple II en 1977) es muy limitada la señal de video que puede generar, y lo único que muestra son números, así que para que esa información sea más entendible, la señal de video final se mezcla con una imagen obtenida de una diapositiva con textos descriptivos. Entonces en la consola se selecciona la diapositiva (hay unas 600 disponibles) utilizando un selector electromecánico y esa selección se pasa a la computadora, que a través de un sistema robotizado en la sala de generación de imágenes toma la diapositiva para mezclarla con una de las salidas de video en particular. La consola luego se limita a elegir que señal de video mostrar. Existen varias señales de video con información, algunas con la posición y velocidad de la nave, otras con parámetros del consumo eléctrico, con datos de la computadora de abordo, también hay con gráficos, por ejemplo el latido del corazón de los tripulantes. Además de todo esto, en la consola existe la posibilidad de capturar una pantalla e imprimirla, pero también es un proceso muy complicado. Al presionar el botón de impresión de la consola, en la sala de imágenes se imprime la pantalla en papel térmico y se la envía a la consola que la solicitó utilizando tubos neumáticos que llegan a cada terminal por debajo.

La electrónica de Houston en las misiones Apolo era extremadamente compleja, con más de 10.000 km de cables y millones de conexiones para unir todas las consolas entre sí, más las enormes unidades de cintas magnéticas para guardar toda la información recibida y transmitida, y grandes impresoras que continuamente imprimían el estado de las naves en formularios continuos.

INYECCIÓN TRANSLUNAR

La maniobra que pone a las naves en dirección a la Luna se llama Inyección Translunar. La tercera etapa del Saturno V enciende su motor y empuja durante 6 minutos, lo que eleva su velocidad unos 3 km/s y modifica la trayectoria para dirigirse hacia la Luna, o, para ser más precisos, hacia donde estará la Luna dentro de tres días (Figura 2). Lo que se logra al aumentar la velocidad en un punto de la órbita circular de 190 km alrededor del planeta, es entrar en una nueva órbita extremadamente elíptica de 190 km en su punto más cercano a la Tierra (perigeo) y de 500.000 km en su punto más lejano (apogeo). Si se acelerase un poco más se podría superar la velocidad de escape de la Tierra (11,2 km/s) y entonces la forma de la trayectoria no sería una elipse (órbita cerrada) sino que se convertiría en una órbita abierta que se aleja por siempre del planeta. Por supuesto, esta nueva órbita elíptica se ve alterada al acercarse al campo gravitatorio de la Luna, así la trayectoria final es una combinación de una órbita elíptica alrededor de la Tierra y otra alrededor de la Luna.

Siguiendo esta trayectoria se estará sobre la Luna en tres días, pero se podría realizar más velozmente si se acelerara más en la inyección translunar, la contra de hacerlo sería que se necesitaría mayor cantidad de combustible en el Saturno y lo mismo haría falta para frenar al llegar a la Luna. Con un empuje menor se puede llegar en un recorrido más largo, pero existe un límite de cuán lento se puede realizar, porque se necesita al menos suficiente velocidad para que las naves lleguen al área de influencia gravitatoria lunar venciendo la gravedad terrestre.

Como la maniobra se trata básicamente de una modificación de la órbita circular, es importante el punto en donde se aumenta la velocidad porque ahí estará el perigeo (punto más cercano al planeta) y en el extremo opuesto el apogeo (punto más alejado) que debe estar, aproximadamente, en el lugar donde se ubicará la Luna en tres días.

Teniendo en cuenta la mecánica orbital, sólo importa la posición de la Luna para determinar donde realizar la inyección translunar, pero existía otro requisito importante y es que esta maniobra vital se tenía que realizar sobre un área de la Tierra con buena cobertura de radio porque no se confiaba en dejar que el Saturno realizara de forma completamente autónoma toda esta maniobra. Los cálculos de cómo debería ser el disparo se realizaban en las computadoras de la NASA y los resultados eran transmitidos a la computadora del Saturno y luego se supervisaba la maniobra de forma exhaustiva para detectar cualquier desviación. Un impulso incorrecto podría poner a la tripulación en una trayectoria de la que, en el peor de los casos, no habría forma de retornar. La zona de la órbita donde había buena

comunicación por radio empezaba en Hawái y abarcaba todo Estados Unidos con una serie de barcos con antenas repetidoras en el medio. Como la Tierra gira sobre su eje, solamente una vez al día coincidía esta zona con el perigeo de la órbita a alcanzar, por eso la maniobra debía realizarse en ese momento en particular, y si no era posible habría que cancelar toda la misión.

A pesar de que toda la maniobra es controlada desde la computadora del Saturno, a la computadora del Módulo de Comando se le cargan los mismos parámetros que se calcularon el Tierra para poder correr un programa que monitoree el correcto desempeño de la maniobra. Este programa muestra en el visor cuanto falta para el momento de la ignición y, una vez que la maniobra comienza, durante el empuje del Saturno muestra en el visor cuanta velocidad (Delta V) falta ganar, para que la tripulación sepa el estado de la maniobra y puedan apagar al Saturno manualmente en caso de detectar un problema.
Cuando se finaliza la inyección translunar, el cohete Saturno (ahora sólo la tercera etapa con muy poco combustible) y las naves vuelan juntos hacia la Luna y lo hacen sin ningún tipo de propulsión, con la velocidad ganada con el último impulso.

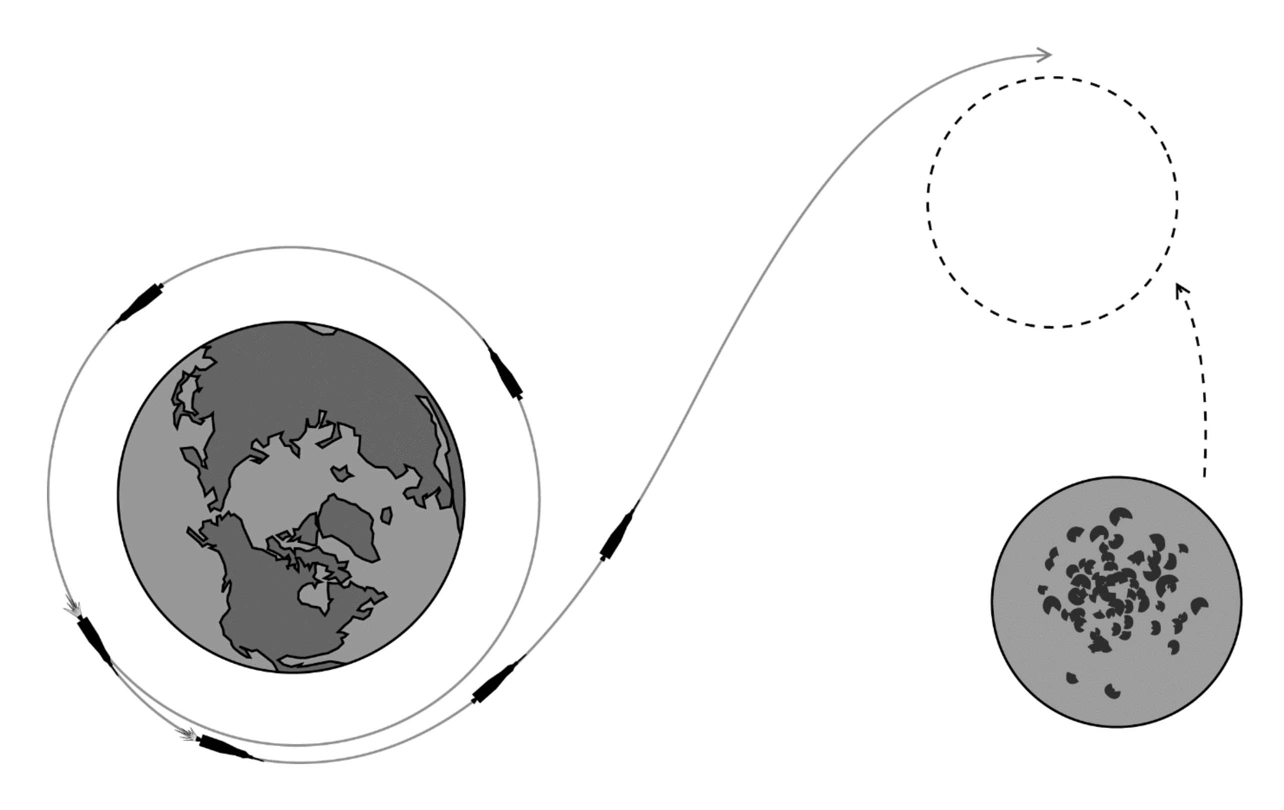


Figura 2. Inyección translunar. Se realiza un cambio de órbita en un punto aproximadamente opuesto a donde estará la Luna al llegar ahí.

Sistemas de navegación de emergencia

Como la tarea que realiza la computadora es vital para la misión, se necesita contar son sistemas de resguardo en caso de fallas. Lo ideal, por simplicidad, sería tener más de una computadora y todas iguales entre sí, entonces si una fallara se utilizaría otra de la misma

manera. El problema es que por su tamaño no era posible llevar más de una computadora como esta, sumado al hecho de que se prefería un método alternativo en caso de que el sistema de navegación en sí no sirviera, por lo que los sistemas usados para emergencia eran totalmente distintos y mucho más simples en su funcionamiento pero más difíciles de utilizar y, por supuesto, los astronautas debían entrenar para el uso de cada uno de ellos.

El más elemental era una serie de giroscopios que indicaban la inclinación de la nave en cada eje y servía para orientarse si no funcionaba la computadora o la unidad inercial. Los astronautas pueden controlar los motores de orientación de forma manual guiándose por estos instrumentos. La contra de estos giroscopios es que no eran tan precisos como la unidad de medición inercial de la computadora, y sólo eran capaces de mantener la orientación por breves periodos de tiempo antes de necesitar recalibrarse.

El otro sistema era el EMS (Entry Monitor System o sistema de monitoreo de reentrada). Este sistema electrónico, como su nombre lo indica, estaba pensado para monitorear la reentrada en la atmósfera, con varias funciones específicas para ese fin, pero servía para algunas otras maniobras simples. Estaba implementado con electrónica digital pero no era una computadora que corría un software sino que sólo cumplía una función. Este instrumento cuenta con un acelerómetro fijo que mide las aceleraciones sobre el eje central de la nave, el eje sobre el que normalmente se acelera. Con la aceleración el circuito electrónico obtiene la velocidad y la distancia desplazada, siempre sobre ese eje, entonces para los casos en que la nave se moviera sólo hacia delante o hacia atrás se puede tomar estos valores como referencia. En el panel del EMS se puede seleccionar si se muestra en un visor de dígitos de siete segmentos, similar al de la computadora, la velocidad o la distancia y se puede modificar manualmente el valor actual para tomar como base cualquier velocidad o posición. Estando en vuelo se descubrió que no operaba bien con el valor 0, por eso se terminó utilizando otro valor como base y se lo restaba mentalmente al observar el visor, porque los valores que muestra son sólo para mirar y ayudar a los tripulantes y no se controla nada con este instrumento.

Despliegue de Módulo Lunar y abandono del Saturno

En viaje hacia la Luna el Saturno ya no cumple ninguna función y es un enorme peso del que es necesario desprenderse, pero primero es necesario extraer el Módulo Lunar que está oculto dentro del cohete. Para poder realizar esta operación, los propulsores de orientación del Saturno giran al cohete y las naves en una dirección pre calculada para evitar ser enceguecidos por el Sol durante las maniobras y luego el comandante presiona un botón que hace detonar una serie de explosivos que cortan la unión entre el Módulo de Comando y el cohete. A partir del momento de la separación del Saturno las únicas computadoras que controlarán la misión son la del Módulo de Comando y la del Módulo Lunar.

La nave se aleja del cohete unos metros utilizando sólo sus pequeños propulsores de orientación porque no hay que ganar mucha velocidad. Como las ventanas del Módulo de Comando apuntan en dirección contraria al Saturno, no hay forma visual de saber cuándo

detenerse, por eso esta maniobra se debe controlar con los instrumentos de navegación. Esto se podría hacer por medio de la computadora pero no existe un programa específico que lo haga y haría falta reorientar la unidad inercial para que pase sobre el eje central de la nave y sea más sencillo de interpretar las coordenadas numéricamente. Lo que se hace es aprovechar que el EMS mide desplazamiento en ese mismo eje y entonces se miden las distancias usando ese instrumento. A una distancia prudencial, se gira la nave y se apunta hacia el cohete, yendo directamente hacia el Módulo Lunar que quedó expuesto una vez que se retiró el Módulo de Comando, y ahora visualmente se realiza la aproximación. El Módulo de Comando se acopla al Módulo Lunar por medio de la escotilla que tiene un mecanismo para este fin y, una vez firmemente unidos, el Módulo de Comando gana velocidad en sentido contrario para extraer al Módulo Lunar de dentro del Saturno. A partir de este momento las naves quedan unidas entre sí pero separadas del cohete y todos en dirección hacia la Luna.

El Saturno sigue viajando junto con las naves a corta distancia, pero es necesario desviarlo para que no interfiera en las próximas maniobras, por eso el propio cohete ejecuta una maniobra de cambio de dirección. En las primeras misiones lo que hacía el cohete era perder velocidad para llegar a la Luna más tarde que las naves, y, como la Luna siguió avanzando durante ese tiempo, el cohete no se acercaba en dirección contraria al movimiento de la Luna (como hacen las naves) sino que entraba por el otro lado y al ser atraído por su gravedad ganaba velocidad, la del movimiento de la Luna, que sumada a la velocidad del Saturno hacía que abandonara el sistema Tierra-Luna y fuera a orbitar el Sol. En las misiones posteriores se desviaba esta etapa del Saturno directo hacia la Luna para colisionar contra su superficie a alta velocidad y de esta forma calibrar los sismógrafos instalados ahí por misiones previas. Todas estas colisiones se hicieron dentro de un rango de 150 Km del punto seleccionado.

Trayecto hacia la Luna

Durante este trayecto el sistema de navegación entra en un modo de *spiedo* (roasting mode) en donde el Módulo de Comando, unido al Módulo Lunar, gira lentamente sobre su eje central. La computadora le aplica una velocidad angular de una revolución por minuto para que el Sol caliente las distintas partes de la nave de forma pareja y no sólo de un lado. Se estudió aplicar un escudo térmico de un lado de la nave para que el lado expuesto al Sol no se calentara de más, pero en el MIT propusieron esta solución, que con unas pocas líneas de programa se ahorran el peso de un escudo [9]. Idealmente, con realizar un disparo inicial de varios propulsores alcanza para dejar a la nave girando sobre su centro de masa sin necesidad de ningún ajuste, pero el rozamiento con los gases internos de la nave hacen que con el tiempo se vaya girando el eje de rotación, por eso es necesario que el piloto automático de la computadora mantenga este modo disparando periódicamente los motores, entre otras cosas para evitar entrar en *gimbal lock*.

Radiación

Durante este trayecto en dirección a la Luna, la nave atraviesa los cinturones de Van Allen, donde una gran cantidad de partículas cargadas eléctricamente orbitan la Tierra atrapadas en su campo magnético. Aquí es el punto de mayor radiación del viaje y los astronautas, aun con los ojos cerrados, pueden ver puntos brillantes que son los choques de partículas con alta energía contra sus retinas. Si bien la radiación en esta región es muy alta, lo es mucho más sobre el Ecuador y el plano orbital de la nave no pasa por ahí, y, además, la nave lo cruza a alta velocidad por lo que no permanecen mucho tiempo aquí.

La computadora de la nave no es sensible a la radiación, ninguna de las memorias lo es y los transistores que utilizan no son afectados por los efectos de largo plazo y de corto plazo que afectan a los semiconductores. En tecnologías más modernas, con un alto grado de miniaturización, una colisión con una partícula cargada puede tener suficiente energía para cambiar el estado de un bit, tanto de una memoria como de un circuito lógico. Aquí los transistores son muy grandes en comparación y necesitan mucha energía para cambiar de estado, ambas cualidades son negativas para casi todo excepto para este caso en particular. Las memorias de soga sólo son alterables de forma mecánica y la de núcleos necesita un campo magnético muy intenso para alterar un bit.

Fuera de los cinturones de Van Allen existe un riesgo de ser atrapados por una tormenta solar, cuando el Sol desprende muchas partículas cargadas en un corto tiempo. El campo magnético terrestre nos protege de estos eventos pero lejos de él sólo la suerte puede proteger a los astronautas. Es un riesgo conocido y aceptado en la planificación de la misión.

Calibración por observación de estrellas

Periódicamente se debe calibrar el sistema de navegación inercial por medio de la observación de las estrellas. En contra de las representaciones que se suelen hacer en la ciencia ficción, en el espacio las estrellas no son todo el tiempo visibles por los ojos humanos y se da la misma situación que en la Tierra: si la iluminación del ambiente es muy intensa, ya sea por el Sol o por luz artificial, no es posible distinguir la tenue luz de las estrellas. Si se está lejos de la Tierra o de la Luna y nada oculta la luz del Sol, hay que orientar la nave de forma tal de mirar por una abertura que apunte en dirección opuesta al Sol para evitar su luz. Incluso así, en el caso de que hubieran partículas que hubieran sido eyectadas por la nave por alguna razón (expulsión de gases para refrigeración, explosiones controladas para desprenderse de partes ya sin uso, o los gases expulsados por los propulsores) cualquiera de estas partículas, por diminutas que fueran, si se encontraran en un ángulo que reflejaran la luz del Sol harían imposible distinguir a las estrellas porque la luz que reflejan es mucho más intensa. El problema es más complejo si se tiene en cuenta que, al no haber una atmósfera que frene a las partículas, estas se moverán junto con la nave hasta que no realice un cambio de velocidad disparando su propulsor. Por eso no es sencillo distinguir estrellas para

hacer las calibraciones, incluso con las condiciones de observación adecuadas el astronauta necesita mirar durante un rato por el visor hasta que sus ojos de acostumbren a su tenue luz y no a la iluminación de dentro de la nave. Además de esto, la visión del espacio exterior desde la nave es muy limitada, no se cuenta con una gran ventana sino varias pequeñas que sólo permiten ver un ángulo muy pequeño del exterior y es necesario girar la nave entera para poder abarcar la totalidad del espacio.

Lo primero que se debe calibrar es la orientación del sistema de navegación porque no tiene sentido calibrar la posición si no se está orientado de forma correcta. No existe la precisión absoluta en la orientación, pero hay un margen de error muy pequeño, de unos pocos segundos de arco, algo así como una milésima de grado. Mientras más lejos se esté de la orientación deseada, más se ira desviando el sistema de navegación con el tiempo al moverse en un sistema de referencia distinto del real. Se seleccionan dos estrellas que sean fáciles de distinguir en la ubicación actual de la nave y se debe apuntar el sextante hacia ellas. La nave se debe reorientar para apuntar hacia cada una de las estrellas. Si la nave ya estaba calibrada, aunque se hubiera desviado ligeramente luego de transcurrido un tiempo, se pueden ubicar las estrellas indicándole a la computadora a cual estrella apuntar. La nave y el sextante se orientarían hacia ella y por el sextante se podría ver la estrella desviada ligeramente del centro del visor. La tarea del astronauta sería hacer el ajuste fino en el sextante para indicarle a la computadora cuál es la orientación de forma más precisa. Si la nave no hubiera estado calibrada, por ejemplo si ocurrió un error y la información del sistema inercial dejó de ser válida, se debe buscar la estrella en todo el cielo, primero usando el telescopio sin aumento del sextante y luego el sextante con aumento, y esto es una operación mucho más larga y lo deseable es sólo ajustar las desviaciones de las calibraciones anteriores.

Una vez que se calibró la orientación es necesario obtener la posición de la nave. Para hacer esto es necesario seleccionar una estrella y medir el ángulo con respecto al horizonte de la Tierra o la Luna. En el sextante se superponen dos imágenes, la del horizonte y la estrella. Se debe girar el sextante hasta que ambas coincidan y en ese momento se debe presionar el botón MARK y se lo debe hacer rápido porque con la velocidad de la nave y la precisión del sextante (un aumento de 28X) esta superposición dura apenas un instante. Cuando la computadora toma la hora exacta en que ambas observaciones coinciden, obtiene la posición de Tierra o Luna (según cual horizonte se observaba) en ese mismo instante y calcula desde que lugar del espacio se tuvo que realizar ese medición para ver ese ángulo en ese momento.

Medición de posición desde Tierra

El sistema de navegación de las naves Apolo se llama sistema de navegación primario, porque para eso fue creado inicialmente. A principios de los ‘60 se pensaba que las naves debían ser totalmente autónomas, no depender de la base en tierra, y si era necesario debían ser capaces de realizar la misión completa sin contacto por radio, entre otras cosas

por miedo a un posible sabotaje de la comunicación a través de interferencias, por supuesto, a cargo de los soviéticos. A medida que fue pasando el tiempo y se empezó a analizar la situación con más realismo, este requerimiento se fue dejando de lado. La hipótesis de que la Unión Soviética quisiera sabotear una misión de exploración espacial se dejó de tomar en serio.

A su vez, la electrónica fue mejorando y la NASA ganó mucha experiencia con los proyectos Mercury y Gemini y con sondas robóticas en cuanto a comunicación por radio y procesamiento de señales. Con la aparición de relojes atómicos extremadamente precisos se hizo posible medir diferencias de tiempo muy pequeñas y así obtener la distancia de una nave a la Tierra por el tiempo que tardaba una señal de radio en ir hacia la nave y ser retransmitida por un transponder de vuelta, todo a la velocidad de la luz. Midiendo el pequeño corrimiento de frecuencia de la señal por el efecto Doppler, se podía obtener la velocidad a la que se desplazaba la nave. La precisión de estas mediciones era muy alta, se podía saber la posición de una nave en la órbita lunar dentro de unos 10 metros, y la velocidad con un margen de error de 0,5 m/s.

La orientación del sistema inercial no se podía obtener desde Tierra sino que debía calibrarse internamente, pero la calibración de la posición no era necesaria realizarla por medio del sextante porque las mediciones por radio eran lo suficientemente precisas. En las primeras misiones se realizaron calibraciones de este tipo con el sextante para poner a prueba el sistema, y se verificó que cumplía perfectamente todos los requerimientos, pero luego se dejó de utilizar por no ser necesarios. Servía para casos de emergencia si se perdía la posición en una ubicación en que no hubiera comunicación por medio de la radio. Desde Houston se le actualizaba periódicamente las coordenadas a la computadora de la nave para que coincidieran con las medidas en Tierra [9].

Ajuste de trayectoria

La inyección translunar que puso a las naves en dirección a la Luna se efectuó con gran precisión, pero si tenemos en cuenta que desde la Tierra la Luna entera representa tan sólo medio grado de arco, una pequeña desviación de apenas unos segundos de arco significa llegar al destino con varios kilómetros de diferencia. A esto se suma que durante el viaje se expulsan gases para refrigerar la nave y también se arrojan residuos, que si bien producen en la dirección de la nave desviaciones muy pequeñas, acumuladas en el tiempo pueden alejarla de su trayectoria.

Para mantener a la nave lo más cerca de la ruta ideal, hay programados a lo largo del viaje siete ajustes de trayectoria. Al llegar a uno de estos puntos se evalúa cuanta es la desviación hasta el momento y se calcula como compensarla. Estos cálculos se hacen desde Tierra con los últimos datos de posición y velocidad de la nave, y así obtienen en que momento tienen que disparar el motor, en qué dirección y cuanta velocidad debe ganar en el empuje. Si se detecta que la trayectoria de la nave está bien encaminada se puede saltear

un ajuste, y es poco probable que se necesiten hacer los siete ajustes en el viaje, pero es casi imposible que no se necesite ninguno en todo el recorrido.

Desde Tierra se le pasa a la tripulación los datos del próximo ajuste de dirección y los cargan en la computadora, que a la hora señalada ejecutará la maniobra tal como fue calculada en tierra. Si el ajuste es muy chico se realiza con los pequeños propulsores usados para ajuste de orientación, sino se enciende el motor principal del Módulo de Servicio. Como la velocidad en sí no se debe modificar sino que se debe ajustar la dirección que lleva, la nave por lo general apunta en dirección perpendicular a su dirección de movimiento para realizar una maniobra de estas y no lo hace apuntando hacia su objetivo.

Abandono del área de influencia gravitatoria terrestre

Estando ya cerca de la Luna se llega a un punto de la trayectoria en que la atracción gravitatoria lunar, por estar tan cerca, es mayor a la terrestre. Hasta ese momento el campo gravitatorio terrestre aceleró la nave hacia la Tierra en sentido contrario a su movimiento, por eso la nave se iba frenando continuamente desde la inyección translunar. A partir de aquí la nave pasa a acelerarse hacia la Luna.

En Houston había monitores que mostraban la posición y velocidad de la nave constantemente, y cuando se llegaba a este punto de la trayectoria cambiaban de sistema de referencia, dejando de medir la posición desde la Tierra y pasando a hacerlo desde la Luna. Los enviados de prensa que no estaban muy interiorizados en los detalles del vuelo espacial reportaban que había habido un salto al ver el brusco cambio en las coordenadas y existía la creencia en la nave sufrían algún tipo de golpe.

Para los astronautas este es un punto del espacio como cualquier otro en la trayectoria. Dentro de la nave, ya sea en la tripulación como en los instrumentos, no existe forma de detectar en lo más mínimo este cambio de dirección de aceleración ya que todo en su interior es afectado de igual manera por los campos gravitatorios. Para saber que ahora se acelera en una dirección y no en otra se puede tomar un punto de referencia externo, como se hacen con las mediciones de posición y velocidad desde Tierra, que obtienen los valores con relación a la posición de las antenas, o por medio del software de la computadora de navegación que calcula la intensidad de los campos gravitatorios en cada punto de la trayectoria y sabe siempre cuánto actualizar la velocidad en base a la aceleración que le corresponde según su posición.

Trayectoria libre de retorno hacia la Tierra

En este momento la nave se aproxima a la Luna cada vez a mayor velocidad porque se ve atraída por su gravedad, pero aun así la trayectoria está calculada para no colisionar directamente sobre ella, sino para pasar de largo. Si no se realiza ninguna maniobra con el motor de la nave, esta deberá ir más allá de la Luna y su gravedad hará que dé una vuelta alrededor de ella, dirigiéndola otra vez en dirección contraria, hacia a la Tierra. Esta

trayectoria es conocida como trayectoria libre hacia la Tierra y está pensada para usar en caso de emergencia por si no encendiera el propulsor principal de la nave para asegurar el retorno de la tripulación a pesar de no haber entrado en órbita lunar.

Inserción en órbita lunar

Como la nave pasa junto a la Luna a una velocidad de 2 km/s, la cual supera a su velocidad orbital, será necesario reducirla en 800 m/s para poder entrar en órbita y no seguir de largo. Si la Luna tuviera una atmósfera se podría usar la fricción con esta para desacelerar, pero en este caso lo que se hace es frenar con el motor del Módulo de Servicio. Se gira la nave hasta quedar con el propulsor apuntando en el sentido opuesto a su movimiento y se dispara durante 6 minutos hasta alcanzar la velocidad requerida. Luego de esta maniobra, la nave queda en una órbita elíptica de mucha excentricidad, pero minutos más tarde, en el punto de la órbita opuesto al último disparo del motor, se lo enciende otra vez durante 16 segundos para quedar en una órbita circular de 110 km de altura, dando una vuelta a la Luna cada 2 horas, una duración mayor a una órbita similar en la Tierra, porque la gravedad, la fuerza que hace orbitar a la nave, es 6 veces menor.

A pesar de estar cerca de la Luna era poco probable tener una buena vista de esta porque una mitad está totalmente a oscuras y además el ángulo de visión desde las ventanas era muy limitado y no se podía reorientar la nave para mejorar la vista al tener el motor empujando en una dirección específica. La tripulación tampoco tenía tiempo de mirar por las ventanas durante este tipo de maniobras, porque debían controlar que todos los procesos relacionados estuvieran correctos y durante las maniobras permanecía cada uno en su asiento para asegurar un determinado centro de gravedad para la nave. Toda la maniobra se hacía sólo por instrumentos, usando la computadora de navegación, y es probable que al entrar en órbita todavía ni hubieran podido divisar la Luna desde el Módulo de Comando durante todo el tiempo transcurrido de la misión.

Esta maniobra para entrar en órbita era vital, si el motor no funcionaba se pasaría a la trayectoria libre de regreso a la Tierra, la misión fracasaría pero se salvaría a la tripulación, aunque en caso de una falla de otro tipo, por ejemplo si el motor se encendiera pero no el tiempo adecuado, las consecuencias serías fatales. Si se reducía la velocidad de más, en poco tiempo la nave se dirigiría directo hacia la Luna y se estrellaría, en cambio, si la velocidad no se reducía lo suficiente no entrarían en órbita pero tampoco seguirían la trayectoria libre hacia la Tierra, sino que se perderían en el espacio. Durante el encendido del motor SPS se controlaba con el visor de la computadora y el EMS que la velocidad ganada fuera la correcta, y se tomaría el control manual del motor en caso de detectarse alguna anomalía.

En gran parte de la órbita alrededor de la Luna no hay comunicación con las estaciones en Tierra porque la nave queda detrás de la Luna que bloquea las ondas de radio. Es justamente en esta zona sin contacto de radio en donde debe realizarse la maniobra para entrar en órbita y no se contará con supervisión del personal en Tierra. De todas formas la información de telemetría que se envía continuamente a Houston, con el estado de varios sensores, en

este caso se grabará en una cinta magnética y luego se transmitirá por otro canal de radio para se pueda analizar el funcionamiento del motor del Módulo de Servicio. Como se conocía bien la trayectoria de la nave, el momento en que se perdía la señal de radio era conocido con anticipación y lo mismo el instante en que volvían a recibir señales, aunque había dos posibilidades, que ocurriera 32 minutos luego de la perdida de comunicación si todo salía bien y entraban en órbita, o sólo 25 minutos más tarde si no entraban en órbita y la nave realizaba la trayectoria libre de regreso. Cuando recibían la señal en Houston ya sabían el resultado de la maniobra, por eso era un alivio que no retomara en el lapso más breve a pesar de no tener ninguna información de la nave durante varios minutos más.

Esta órbita no es muy estable en el tiempo porque se descubrió que la masa de la Luna no está repartida de forma muy homogénea y hay concentraciones de masa, conocidas como Mascons gravitatorios (de Mass Concentrations), que alteran el movimiento orbital y la nave necesitará periódicamente ajustar la trayectoria.

Separación del Módulo Lunar

Hasta este momento los tres astronautas permanecieron juntos en el Módulo de Comando durante el transcurso de la misión, pero ahora que ya están orbitando la Luna se deberán dividir en dos grupos, el comandante y el piloto del Módulo Lunar se pasarán a esta última nave para realizar el descenso a la Luna mientras en el Módulo de Comando quedará su piloto orbitando solo, quien a pesar de hacer todo el viaje hacia la Luna no participará de la exploración de la superficie.

Antes de separar las dos naves, unidas por una escotilla, deberán entrar al Módulo Lunar y verificar que todo esté en correcto estado. Esta nave permaneció cerrada y con todo el instrumental apagado para ahorrar oxígeno de sus tanques y electricidad de sus baterías. Se encienden todos los sistemas eléctricos y se verifican que estos funcionen correctamente, entre ellos la computadora de navegación. Antes de probar los motores, se detona una carga explosiva que permite desplegar las patas que servirán para apoyarse sobre la Luna y que estaban contraídas para poder caber dentro del Saturno, y también se hace explotar otra carga que abre la válvula de gas helio para darle presión al combustible lo que permite que pueda ir a los motores cuando se los active. Con la nave en la configuración lista para volar, se realizan pruebas de disparo de los distintos motores para verificar que todo esté en orden para utilizar esta nave.

Ahora la computadora del Módulo Lunar debe configurarse porque hasta el momento nunca se usó en la misión. La mayoría de la información a ingresársele también está en la computadora del Módulo de Comando pero no existe ningún tipo de interconexión entre los dos equipos, por eso la carga de los parámetros del sistema se debe realizar a mano mientras otro astronauta dicta lo que lee en la computadora del Módulo de Comando. Lo primero a cargarle en la computadora es la hora, que debe ser la misma para las dos naves y para Houston. Luego se le cargan la orientación y coordenadas de la computadora de navegación de la otra nave para que oriente la plataforma inercial en la

misma dirección y se ubique en la misma posición en el espacio. Estos tres ángulos con la orientación, antes de ser ingresados en la computadora, deben ser transformados teniendo en cuenta la inclinación entre ambas naves. La unidad inercial de cada nave mide los ángulos con respecto a la nave en sí pero no está orientada de la misma manera en cada una, sino que en el Módulo de Comando tiene una inclinación y en el Módulo Lunar otra, porque este instrumento está unido al visor de usado para observar las estrellas, que en el Módulo de Comando tiene una inclinación de 32 grados con respecto al cuerpo de la nave. Es necesario convertir los datos leídos de una nave al sistema de medición de ángulos de la otra para que ambas unidades inerciales apunten en la misma dirección y las dos naves se puedan orientar en el mismo sistema de referencia. Aun haciendo ese ajuste hay tener en cuenta en la conversión de los ángulos que el pasillo que une al Módulo Lunar con el Módulo de Comando no es completamente rígido y existe una inclinación en esta unión y por eso este tiene un indicador de ángulos para tomar en consideración al hacer la transformación. Esta cuenta para transformar los ángulos leídos de una computadora a la otra se debe realizar a mano porque no se justificaba el esfuerzo de escribir un programa para hacer algo que se debe realizar una sola vez en la misión. De todas formas, esta orientación cargada en la computadora no tiene la precisión necesaria para realizar un viaje tan largo sin desviarse significativamente, es sólo una aproximación, por eso luego se deberá realizar un ajuste fino utilizando el instrumental propio del Módulo Lunar.

Cuando está todo listo se cierra la escotilla y se destraban los soportes del Módulo Lunar para que se separe del Módulo de Comando. Esta unión entre las naves está diseñada para usarse varias veces durante la misión, cuando recién se acoplan al descartar al Saturno V, cuando se separan para alunizar, luego cuando regresan de la Luna y finalmente cuando se descarta al Módulo Lunar, por eso no utilizan explosivos como en los otros casos en que se separan de secciones de la nave o de partes del cohete. Este mecanismo de unión es complejo y confiable. Una vez que ya no están unidas las naves, el Módulo Lunar gana un poco de velocidad para alejarse del Módulo de Comando por medio de disparos de sus propulsores pequeños y recién a una distancia prudencial puede encender el motor principal sin correr el riesgo de afectar a la otra nave.

Orientación del Módulo Lunar

La alineación precisa en el Módulo de Comando se realiza utilizando el sextante controlado por la computadora, pero este instrumento es grande y pesado y no sería posible tener uno igual en el Módulo Lunar porque esta nave debe ser muy liviana y pequeña y, además, a diferencia del Módulo de Comando por sí sola no realiza un viaje tan largo de cientos de miles de kilómetros, por eso no necesita la funcionalidad de obtener su posición en el espacio por medio de observaciones a las estrellas y planetas sino que le alcanza con obtener la posición de la otra nave o incluso la transmitida desde Houston. Lo único que se necesita es un instrumento para orientarse en base a las estrellas, para ajustar con precisión la unidad inercial y hacer que apunte en una dirección precisa en el espacio.

El instrumento del que dispone la nave es el *telescopio de alineación óptica* que es un telescopio que apunta hacia arriba de la nave y tiene visión sin aumento con un ángulo de 60 grados y se puede apuntar, de forma manual sin ningún motor, en sólo 6 direcciones fijas a 60 grados de diferencia cada una [10]. Existían dos formas de usarla para orientarse, una en vuelo cuando la nave se podía controlar libremente y la otra cuando estaba unida al Módulo de Comando o estaba en la superficie lunar y no había posibilidad de girar.

En vuelo libre se seleccionaba una estrella (las estrellas que se ven en la Luna o sus alrededores siguen siendo las mismas que se ven desde la Tierra) y se le indica a la computadora que estrella es esa. El sextante del Módulo de Comando se puede apuntar moviéndolo fracciones de grado en cualquier dirección y la computadora, que está integrada con este instrumento, sabe con precisión estos ángulos, en cambio aquí sólo se puede apuntar en seis posiciones fijas y se le indica a la computadora el código que identifica en cuál de las seis posiciones está apuntando para que lo tenga en cuenta en los cálculos. Luego se hace rotar la nave en un eje y se mira por el visor del telescopio esperando que la estrella coincida con una de dos rayas perpendiculares marcadas en el visor, una es la X y la otra la Y. En el momento en que la estrella toca esa línea se presiona uno de dos botones (Mark X y Mark Y) en el telescopio, y como estos botones están conectados directamente a la computadora, así se le indica que en un eje en particular se está apuntando directo hacía la estrella seleccionada. Se debe realizar el mismo procedimiento rotando en dos ejes perpendiculares entre sí y con estas dos mediciones la computadora conoce la orientación de esta estrella con respecto a la unidad inercial. Se realiza esto mismo con dos estrellas distintas y de esta forma la computadora calcula su orientación en el espacio y rota a la unidad inercial para que apunte en la dirección correcta según el sistema de referencia en uso.

La orientación cuando la nave no se puede mover libremente se hace utilizando otra serie de marcas en el visor del telescopio, y también es un proceso realizado en dos pasos para cada una de las dos estrellas a medir, pero en lugar de tomar mediciones mientras la nave rota, se lo hace girando el visor, que tiene un mecanismo para girarlo manualmente y un medidor mecánico de ángulos con una precisión de 2 décimas de grado, y se puede ver el ángulo de giro del visor en 4 dígitos. Existen dos tipos de marcas que se deben comparar con la estrellas a medir. La primera son dos líneas que parten desde el centro del visor y están separadas por un ángulo fijo. El astronauta debe girar el visor hasta que la estrella quede ubicada en el centro de estas dos líneas, ahí toma el número del medidor del ángulo del visor y lo ingresa en la computadora y este es el ángulo en que se encuentra la estrella con respecto al centro del visor. Luego se rota al visor pero observando los líneas curvas en espiral que también parten desde el centro del visor, y cuando la estrella quede en el medio de estas dos líneas se debe tomar el ángulo e ingresarlo a la computadora: este número le sirve para, junto con la medición anterior, calcular la distancia de la estrella desde el centro del visor. Con estos dos valores obtiene la dirección en que se encuentra la estrella, y al medir dos estrellas se orienta toda la unidad inercial.

Computadora de Backup del Módulo Lunar

El Módulo Lunar cuenta con una computadora de backup en caso de que falle el sistema principal, pero igual que ocurre en el Módulo de Comando, la computadora de backup es completamente distinta a la principal y mucho más simple y limitada. El AGS (Abort Guidance System, sistema de guiado para abortar) como su nombre lo indica estaba pensada específicamente para poder abortar la misión si se rompía el sistema de guiado principal, no para continuar el descenso sino para poder poner a salvo la tripulación [1]. Es un sistema completo con computadora y unidad inercial para guiado de la nave y es un desarrollo independiente al del MIT, en este caso realizado por la empresa TRW dedicada a la industria aeroespacial y militar.

Las características de esta computadora son muy distintas a la principal: es de 18 bits, utiliza números con el sistema de complemento a 2 en lugar de complemento a 1, y la memoria es sólo de núcleos magnéticos y no tiene memoria ROM de ningún tipo. Su sistema inercial también es distinto, y en lugar de usar giroscopios montados en una plataforma independiente usa giroscopios individuales, cada uno con un acelerómetro, un sistema menos preciso y sólo útil en lapsos breves porque necesita calibrarse más seguido.

La interfaz de usuario del sistema de emergencia era incluso más básica y difícil de utilizar que la de la computadora principal. Cuenta con un solo visor de 5 dígitos y un teclado con menos teclas. La secuencia de utilización era también más simple desde el punto de vista del programador pero mucho más complicada de utilizar, porque se debía ingresar la información de forma más técnica, indicando la dirección de memoria donde se encontraba el dato, similar al debuger de un microprocesador. Por supuesto, al acceder directo a cualquier posición de memoria, era más fácil realizar una operación errónea que desconfigurara todo el sistema y había que chequear con mucho cuidado lo que se escribía.

Durante toda la operación del Módulo Lunar, el piloto revisa el funcionamiento del AGS para verificar que tenga la misma información que el sistema de navegación principal, si detecta una desviación, actualiza al sistema secundario con la información de la computadora principal que es más confiable. El piloto dedicaba mucho de su tiempo a operar esta computadora por lo complicado de su interfaz. Si ocurriera un problema con el sistema de navegación principal, ya sea porque dejó de funcionar o se desalineó y no hay tiempo de corregirlo, se pasa el control al AGS, se eyecta la etapa de descenso del Módulo Lunar y se asciende hacia el Módulo de Comando usando este sistema de navegación de emergencia.

INSERCIÓN EN ÓRBITA DE DESCENSO

Una vez que el Módulo Lunar es independiente del Módulo de Comando, debe abandonar la órbita circular de 110 km de altura, que comparten ambas naves, y pasar a una órbita elíptica con un apolunio (punto más alejado de la Luna) de todavía 110 km pero un perilunio (punto más cercano a la Luna) de 16 km de altura. Esta es la órbita de descenso

porque, en un punto, pasa a una distancia lo suficientemente cercana a la superficie para iniciar el descenso. En la Tierra la altura mínima para entrar en órbita es donde la atmósfera ya es lo suficientemente tenue para no frenar la nave, pero, como en la Luna no hay atmósfera, al momento de diseñar la misión se eligió una altura mínima de 16 km como un margen de seguridad para que la nave no chocara con ninguna montaña, porque no había un conocimiento muy detallado de la superficie y se supuso que a esta altura no debía haber ningún pico elevado de la Luna. También se tuvo en cuenta que, en caso de ocurrir un problema de navegación y se terminara en una órbita unos kilómetros más baja, tampoco hubiera riego de choques mientras todavía se orbita la Luna, antes de iniciar el descenso.

Como el perilunio debe estar ubicado en un punto en particular que permita iniciar el descenso desde ahí y tener comunicación con radio directo a la Tierra (para poder supervisar todo el descenso), esta maniobra se debe realizar desde el punto opuesto de la órbita (el apolunio) en la cara oculta de la Luna y sin comunicación por radio, de forma similar a la maniobra realizada por el Módulo de Comando para entrar en órbita alrededor de la Luna, pero esta vez se realiza por medio de la computadora del Módulo Lunar y con una duración de unos 30 segundos.

Inicio de maniobra de descenso: frenado

Para lograr alunizar, el Módulo Lunar debe pasar de orbitar la Luna a una distancia mínima de 16 km con una velocidad de 6.000 km/h respecto al suelo, a posarse suavemente sobre un punto en la superficie [1, 9]. No existe nada externo que frene a la nave, así que esta deberá reducir su velocidad utilizando su motor y toneladas de combustible. Estando a alturas elevadas se trata básicamente de permanecer frenando continuamente, pero cuando ya se está a baja velocidad y cerca del suelo las maniobras pasan a ser muy distintas: la nave en este caso trata de compensar la atracción gravitatoria para descender casi verticalmente sobre el punto designado. Toda esta secuencia es la tarea más complicada que debe ejecutar la computadora porque no es vuelo inercial con unos pocos disparos del motor en una dirección en particular sino que durante 11 minutos se debe controlar continuamente el propulsor en distintas etapas.

El sistema de coordenadas utilizado en el descenso tiene como inicio el centro de la Luna y uno de los ejes de coordenadas pasa por el punto prefijado de alunizaje y otro de forma paralela al plano en que se mueve la nave, de esta forma, al llegar el destino, idealmente dos de las coordenadas serán 0 y la otra la distancia de ese punto al centro de la Luna, lo cual hace más simple la interpretación de las coordenadas durante todo el descenso.

El motor de descenso del Módulo Lunar es capaz de regular la potencia que entrega, controlando cuánto tiene que empujar en cada momento, a diferencia de los otros motores usados en las naves que tienen una potencia fija y se los controla solamente administrando la duración en que permanecen encendidos. Este motor de descenso, en cambio, en el frenado se enciende de forma continua y pero, además, no sólo se usa para modificar la

velocidad en órbita, sino que también se usa cerca del suelo para contrarrestar la aceleración de la gravedad cuando la nave se mueve en dirección vertical, y aquí se necesita menos potencia y debe variar según la cercanía a la Luna y también de acuerdo al peso de la nave, que baja todo el tiempo por su constante consumo de combustible. Una complicación en el control del motor de descenso era que solamente se lo podía utilizar entre 0 % y 55 % de la potencia, y sino saltar directamente al 100 %, porque en el otro rango se desgastaba mucho. De hecho siempre que se utiliza se va desgastando a causa de los químicos corrosivos que usa, pero esta pérdida de potencia está estimada de antemano y se tiene en cuenta en el software que ejecuta las maniobras.

El descenso a la Luna se divide en tres fases, la primera es la de frenado, que es la encargada de reducir la mayor parte de la velocidad y es, básicamente, una larga maniobra orbital para llevar a la nave a una órbita muy baja, que de hecho es tan baja que no es posible completarla porque la nave chocaría contra la superficie si no se pasara a la siguiente fase de descenso a tiempo. La maniobra de frenado es ejecutada por la computadora utilizando el programa 63 y se lo carga antes de llegar al punto más bajo de la órbita (16 km de altura), y el programa, luego de aceptar con la tecla PRO varias confirmaciones de parámetros de la trayectoria, hace girar la nave para que su motor apunte en dirección contraria a su movimiento orbital alrededor de la Luna. A continuación muestra en el visor el tiempo restante para iniciar el frenado hasta que, faltando 5 segundos para encender el motor, los tripulantes deben presionar PRO para iniciar el frenado. Pasado este tiempo deberán esperar otra órbita completa para volver a intentarlo, pero nunca el software por sí solo iniciará la maniobra. El frenado comienza al encenderse el motor, pero, durante el primer medio minuto, lo hace a sólo un 10 % de su potencia para que la computadora calibre la tobera, verificando que el empuje pase sobre el centro de masa y que no haga rotar a la nave. Luego de ajustar su dirección, el motor se enciende al 100 % y, durante 9 minutos, se frena a la nave que continuamente va perdiendo velocidad y describe una trayectoria cada vez más cerca de la superficie (Figura 3.a).

El comandante es el encargado de mirar por la ventana, analizando la trayectoria, porque en esta etapa de frenado es todo automático, solamente busca distinguir en la superficie (todavía a varios km de distancia) detalles de la geografía lunar para verificar que se esté bien orientado. El piloto del Módulo Lunar que, a pesar del título que posee, cumple el rol de copiloto, continuamente monitorea los instrumentos y los distintos visores de las dos computadoras, la de navegación y la de backup. El comandante nunca mira los instrumentos ni opera la computadora, el piloto es el encargado de esto y le dice al comandante la información de los instrumentos, como la altura y la velocidad vertical, para que este se ocupe de controlar el descenso.

Radar de aproximación

Estando a aproximadamente 10 km de altura sobre la Luna, se debe activar el radar para medir la distancia y velocidad de descenso hacia la superficie para dejar de navegar

solamente en base a la unidad inercial y tomar una referencia externa (Figura 3.b). La mejor información sobre la altura del punto de descenso se obtuvo en la misma misión en observaciones realizadas en órbitas previas desde Módulo de Comando, que, por medio del sextante controlado por la computadora, midió con precisión el ángulo del destino desde dos puntos de la órbita. Con esa información en Houston calculaban la altura a la que debía estar con respecto al centro de la Luna antes de mandarle los datos para la maniobra de descenso. De todas formas, la última palabra en cuando a posición con respecto a la superficie lunar la dará el radar con su medición directa de esta distancia.

La etapa de frenado se realiza con la nave orientada de forma horizontal con respecto a la superficie, inicialmente con las ventanillas mirando hacia la Luna, pero luego es necesario girar la nave de forma tal que el motor siga apuntando en la misma dirección y la nave siga horizontal pero con las ventanillas mirando hacia el espacio. Esto es así para que el radar de aproximación, ubicado debajo de la nave, pueda apuntar hacia la superficie en una de sus dos posiciones posibles. La otra posición es directamente hacia abajo y se usa así cuando la nave pasa a descender de forma vertical.

Cuando el radar comienza a enviar información se puede detectar mirando el tablero, porque la luz *ALTITUDE* se apaga al recibir información de la distancia desde el radar, y luego, cuando ya genera información de velocidad de descenso, se apaga la luz *VELOCITY*. Se ejecuta un comando en la computadora para mostrar la diferencia entre la altura calculada por el sistema de navegación y la del radar, y la computadora toma la del radar como información válida para el descenso, pero, si esta diferencia fuera muy alta, el comandante debería abortar el descenso porque se estaría trabajando con información poco confiable, ya sea del radar o del sistema de navegación.

El punto de destino en la superficie se seleccionaba para que la trayectoria de los últimos Km, cuando se activaba el radar, pasara por zonas llanas y así no obtener información muy variable en el radar en la aproximación. Lo ideal es que lo leído correspondiera a lo que faltaba descender y no a un cráter por el que justo se sobrevolaba, porque el sistema de navegación lo podría interpretar como que había que descender mucho más de lo que realmente había que hacerlo.

Continuación de maniobra de descenso: aproximación

La etapa de frenado termina en un punto conocido como *High Gate*, ubicado a 2 km de altura y unos 7 km de distancia horizontal al punto de alunizaje (Figura 3.c). Al comienzo del frenado el Módulo Lunar pesaba unas 16 toneladas, al finalizar ya perdió casi la mitad de su peso luego de consumir gran parte de su combustible para reducir la velocidad hasta unos 600 km/h. Cuando la computadora detecta que se llegó a *High Gate* automáticamente pasa al programa 64, que es el encargado de comandar la nueva fase del descenso, la de aproximación. Hasta el momento la trayectoria fue totalmente automática y comandada por la computadora sin posibilidad de modificar el destino, que de hecho los tripulantes realizaron sin ver exactamente hacia donde se dirigían. La misión fue planificada usando

fotografías tomadas en órbita por sondas robóticas o misiones Apolo anteriores, pero a partir de este momento se está tan cerca de la Luna que se puede ver la superficie con mayor detalle del que se disponía en la planificación y los astronautas pueden tomar decisiones en base a lo que observan. La computadora gira la nave, que queda en posición casi vertical, con el motor apuntando hacia abajo, lo que permite a la tripulación mirar hacia el punto de la superficie al que se dirigen a través de las ventanas que tienen un ángulo especial para que se pueda mirar hacia el suelo estando de pie (de hecho el Módulo Lunar no tiene asientos para ahorrar peso).

Esta etapa dura en total unos 80 segundos y, mientras se sigue moviendo horizontalmente y se continúa frenando, el comandante puede seleccionar otro lugar de alunizaje, para no descender sobre una zona muy irregular o dentro de un cráter. Para seleccionar el destino se utiliza la computadora porque el vuelo sigue siendo totalmente automático controlado por software. En el visor de la computadora se muestra en esta etapa un valor en grados que es el ángulo del *Designador de Punto de Alunizaje* (LPD, Landing Point Designator). Para saber hacia donde los está dirigiendo la computadora se mira a través de la ventana tomando en cuenta una grilla que tiene dibujada, tanto del lado de afuera del vidrio como del lado interior. El comandante se ubica en una posición tal que estas dos grillas se superponen, desde ese lugar mirando los números gravados junto a la grilla el astronauta identifica el ángulo del *designador* y así ve cuál es el punto en la superficie hacia donde se dirigen. La solución natural, siguiendo el diseño del Módulo de Comando, hubiera sido incluir un pesado visor controlado por la computadora para poder ver y seleccionar el objetivo del sistema de navegación, pero aquí, para ahorrar peso, se resolvió con unas simples líneas gravadas en la ventana y unas rutinas de software que, sabiendo la geometría del Módulo Lunar, la ubicación desde donde se debía mirar por la ventana y la orientación de la nave lo resuelve todo mostrando un solo número.

En caso de querer cambiar el lugar de alunizaje, siempre dentro de un eje, el comandante puede orientar la nave hasta encontrar un lugar propicio, y luego mira por la ventana usando el indicador y le dice al piloto el ángulo que ve en la grilla. El piloto ingresa este valor en la computadora y esta cambia el punto de destino y recalcula la trayectoria. El software también soporta una forma más sencilla de cambiar el punto de alunizaje, simplemente dándole un pequeño movimiento a la palanca de mando, lo que, en vez de rotar un poco la nave, subiría o bajaría medio grado el ángulo de inclinación hacia dónde dirigirse. Ese cambio de ángulo podía significar una distancia de varios cientos de metros o sólo algunas decenas, según la altura a la que se encontrasen. Mientras más alto se esté es mayor el rango de lugares a donde puede seleccionar el destino, a medida que se baja se van acotando los posibles lugares de descenso.

Esta etapa termina en un punto conocido como *Low Gate* a 200 metros de altura y 600 metros horizontales hacia el punto de alunizaje.

Final de maniobra de descenso

En *Low Gate* se pasa a la última fase, la de descenso, donde la nave realiza una corta trayectoria casi vertical de 200 metros hacia la superficie, controlando el movimiento horizontal para seleccionar el punto exacto donde tocar la Luna (Figura 3.d). La computada solicita confirmación del comandante para decidir cómo continuar. Existen varias alternativas además de, por supuesto, abortar el descenso. Se puede optar por continuar con el programa automático (el 65) que realiza el descenso sin seguir ninguna instrucción del comandante, o utilizar un programa de descenso semiautomático (el 66) en donde la computadora sólo controla la velocidad de descenso y el comandante por medio de la palanca controla la velocidad horizontal. También existía otra opción que era un programa totalmente manual (el 67) en donde el comandante controla de forma total tanto la velocidad horizontal como la vertical, aunque era muy riesgoso porque un mal control de la velocidad de descenso podía hacer que en poco tiempo la nave ganara mucha velocidad de caída y se estrellara contra la superficie, además de ser poco óptimo en el uso del poco combustible restante comparado con el software dedicado. En la práctica siempre se utilizó el 66 semiautomático para que el comandante controlara a ojo el lugar donde alunizar.

La nave queda totalmente vertical y el comandante la puede mover en cualquier dirección horizontal para dirigirla a un punto en particular, porque ahora que está a apenas metros de la superficie se pueden ver detalles pequeños que pueden interferir con un buen alunizaje. Esta etapa también dura aproximadamente 80 segundos pero se puede alargar según el combustible que se posea si el comandante, en vuelo semiautomático, mueve la palanca de velocidad vertical para indicarle a la computadora que no descienda tan rápido para darle más tiempo de seleccionar un punto de alunizaje. El piloto continuamente lee cuanto combustible hay disponible para decidir si se puede terminar el descenso o hay que abortar.

Se desciende verticalmente los últimos metros antes de tocar la superficie y para ese momento la velocidad horizontal debe ser casi nula, porque la nave podría rotar al tocar con las patas el suelo si tiene todavía velocidad lateral. Como a muy corta distancia la información del radar no es precisa, estando a apenas centímetros de la superficie unos interruptores (llamados las sondas de contacto) presentes en 3 de las 4 patas del Módulo Lunar, se activan e indican, por medio de una luz azul en el panel, que se está casi tocando la superficie. Es responsabilidad del comandante detener el motor en este momento porque el software, como en la mayoría de las decisiones importantes, no lo hace automáticamente. Si el comandante tarda más de unos pocos segundos en apagarlo, se corre el riesgo de que la potencia del motor reflejada sobre el suelo dañe a la propia nave. Si, en cambio, se apura mucho se puede perder sustentación estando a una altura que rompería a la nave al golpear fuerte contra el suelo. Al apagar el motor se toca la superficie a una velocidad de un 1 m/s, con una nave que pesa unas 8 toneladas, pero como la gravedad es 6 veces menor a la de la Tierra y las patas del Módulo Lunar amortiguan la caída, no se producen daños en este contacto.

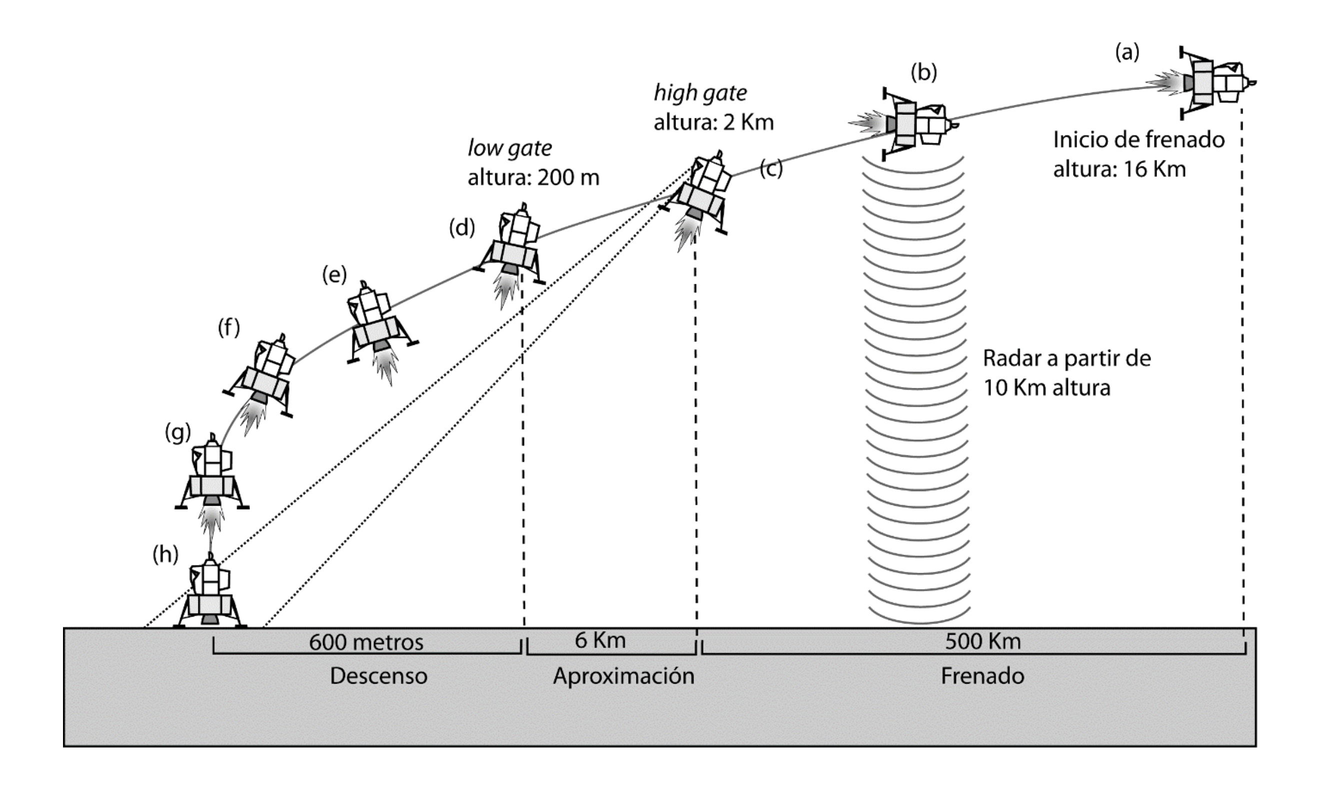


Figura 3. Descenso a la Luna de Este a Oeste según las coordenadas lunares. (a) La parte más larga es el frenado, sin visibilidad hacia la superficie. (b) El radar comienza a medir la altura y velocidad de descenso. (c) La nave gira y durante esta etapa se puede ver el destino al que el software la dirige y se permite modificarlo. (d) Se comienza el descenso directo. (e) y (f) Según la orientación de la nave se modifica la velocidad horizontal mientras el software regula la potencia para que el empuje hacia arriba contrarreste a la gravedad. (g) Habiendo anulado la velocidad horizontal se desciende de forma totalmente vertical. (h) Se culmina el descenso apoyando las patas en la superficie.

EXPLORACIÓN DE LA SUPERFICIE LUNAR

Una vez que el Módulo Lunar queda posado sobre la superficie, los astronautas se preparan para realizar sus expediciones y la computadora en esta etapa no cumple ninguna función. Los tripulantes apagan varios sistemas eléctricos innecesarios para ahorrar batería, entre ellos la computadora, aunque en las primeras misiones, que eran cortas, se la podía dejar encendida para que estuviera orientada en caso de necesitar partir sin calibrar el sistema de navegación. En algunos casos se realizaron experimentos desde la Tierra controlando la computadora remotamente para medir con precisión la gravedad lunar utilizando la plataforma inercial. Se la hacía girar y se obtenían las distintas mediciones de los acelerómetros.

Cuando los astronautas salen a caminar sobre la Luna lo hacen protegidos del vacío en unos muy complejos trajes espaciales presurizados que incluyen, entre otras cosas, control de temperatura y una radio para comunicarse entre ellos y con la nave, que retransmite la señal a la Tierra, pero no llevan consigo ningún sistema de navegación ni computadora de ningún tipo. Todo el recorrido a pie debe realizarse cuidadosamente en el

paisaje monótono de la Luna tratando de no perder la referencia de donde está el Módulo Lunar para luego poder volver.

En las últimas misiones ya contaban con un vehículo eléctrico de cuatro ruedas para poder explorar un área mucho mayor que la se podía hacer a pie. Como los recorridos con este vehículo eran relativamente grandes, este contaba con un sistema de navegación electrónico, pero mucho más simple que el usado en las naves. Podían ver en cada punto de su recorrido las coordenadas con respecto al lugar de alunizaje, pero este sistema no tenía ninguna interfaz de entrada porque esto era lo único que realizaba, no hacía falta seleccionar otro programa ni ingresar parámetros. Por las dudas de que este sistema de navegación no funcionara correctamente y a falta de brújulas (no funcionan en la Luna al no tener un campo magnético) se contaba con un reloj de Sol en el panel, cuya sombra ayudaba a saber la orientación en la superficie, aprovechando que el Sol se mueve muy lentamente en el cielo lunar y la sombra apunta durante días en la misma dirección.

DESPEGUE DESDE LA SUPERFICIE DE LA LUNA

Al terminar la exploración, los astronautas deben volver al Módulo Lunar para abandonar la superficie y reencontrarse con el Módulo de Comando en órbita y con él volver a la Tierra. La parte inferior del Módulo Lunar, la sección que tiene las patas y el motor de descenso con su tanque de combustible (ahora vacío), ya no cumple ninguna función, y la nave deberá dividirse en dos y abandonar esta mitad en la Luna. La mitad superior del Módulo Lunar, la etapa de ascenso, posee otro motor con su propio tanque de combustible y es con este propulsor que se despegará. La gravedad lunar es muy tenue comparada con la terrestre y, además, no hay una atmósfera que ofrezca resistencia al movimiento, por eso no es necesario un cohete muy poderoso ni tanto combustible. La etapa de ascenso es muy pequeña y liviana y, para reducir aún más su peso, se abandona en la superficie lunar todo lo que ya no sea necesario, como el sistema de refrigeración portátil de cada astronauta y sus tanques de oxígeno.

La nave no puede partir en cualquier momento sino que debe despegar cuando el Módulo de Comando pase por un punto en particular de su órbita para que luego de varias maniobras orbitales puedan encontrarse en un mismo lugar. El piloto del Módulo de Comando orbitó en soledad la Luna mientras los otros astronautas descendían y exploraban la Luna, ahora queda en espera de que vuelvan sus compañeros, pero para que se puedan encontrar en órbita es importante que tanto el Módulo Lunar y el Módulo de Comando orbiten en el mismo plano. Al momento de separarse ambos estaban en un mismo plano, pero mientras el Módulo Lunar permaneció un tiempo largo (horas o días) en la superficie lunar, durante ese tiempo la Luna rotó sobre su eje, si bien no una vuelta completa, pero sí lo suficiente para apartar al Módulo Lunar del plano original. En las primeras misiones, de corta duración, se mantenía esta diferencia de planos orbitales y se dejaba que el Módulo Lunar luego ajustara su trayectoria, pero en las misiones posteriores, que eran más largas y por eso tenían diferencias de planos más pronunciadas, el propio Módulo de Comando es el

que realiza la maniobra para cambiar a un nuevo plano orbital que pase por donde se encuentra el Módulo Lunar en la superficie, así al momento de despegar el Módulo Lunar y el Módulo de Comando ya estaban en el mismo plano y se simplificaban las maniobras posteriores.

En el Módulo Lunar deben encender la computadora para controlar el despegue pero antes de realizar cualquier maniobra deben calibrar su sistema de navegación. Para la reorientación alcanza con apuntar a sólo una estrella, el otro eje de alineación se obtiene por medio de los acelerómetros que se orientan en base a la fuerza de gravedad al estar la nave sujeta a la superficie y se usa un programa pensado para este caso. Luego era necesario cargarle las coordenadas de la nave en el espacio, pero el Módulo Lunar no tiene instrumental para obtener esta información, aunque el lugar donde se encuentra es el mismo sobre el que alunizó y las coordenadas en el espacio sólo se alteraron por el movimiento de la propia Luna. Para obtener una mejor estimación de la posición del Módulo Lunar se lo puede localizar desde el Módulo de Comando aprovechando el momento de la órbita en que lo sobrevuela y se apunta el sextante en la dirección en que debería estar. A 110 km de altura es imposible observar algo tan pequeño, pero con un aumento de 28X si se cuenta con una estimación previa de la posición y se apunta en la dirección correcta se lo puede distinguir. Si se lo encuentra, la computadora calcula las coordenadas con gran precisión en base al ángulo del sextante y las coordenadas del Módulo de Comando. Si no se lo pudo ubicar igual se puede continuar normalmente porque se tiene una estimación de la posición, en el peor de los casos la nave se pudo desviar algunos kilómetros. Además, la maniobra de encuentro entre las naves está pensada para poder realizarse con la información que obtiene el Módulo Lunar por medio del radar de encuentro.

Se obtienen los parámetros para el despegue desde Houston y se los carga a la computadora para que a la hora adecuada comience la maniobra. En el visor se muestra una cuenta regresiva que culminará en 0 cuando la nave encienda el motor y ascienda. Faltando 5 segundos el programa muestra 99 en el visor solicitando confirmación del comandante que debe presionar PROCEED, de forma contraría el software no podrá encender el motor por su cuenta. Antes el comandante también debió presionar un botón, independiente de la computadora, que separa al Módulo Lunar de la etapa de descenso, detonando unas cargas explosivas que cortan todas las uniones, tanto soportes como cañerías y cableado.

En un ascenso en un cohete desde la Tierra se parte de forma casi vertical y luego de varios minutos, al abandonar la parte más densa de la atmósfera, se profundiza el giro en dirección horizontal. Como aquí no hay atmósfera, la nave apenas despega gira 50 grados para ganar velocidad horizontal y realiza el ascenso de forma diagonal (Figura 4). Para ahorrar peso, el motor de esta etapa de ascenso no cuenta con una tobera direccionable, sino que siempre empuja de forma fija sobre el mismo eje, por eso es importante ubicar todas las cosas en su lugar dentro de la nave para tener el centro de gravedad bien alineado. La computadora ajusta la dirección cada dos segundos disparando los propulsores de alineación que hacen girar toda la nave para que siga la trayectoria prefijada, por eso mientras más desalineado esté el centro de gravedad de la nave, más la habrá que

compensar con disparos de los propulsores de alineación, gastando más combustible. Al partir la nave sufre una aceleración hacia arriba de 1/3 G, el doble de lo que los astronautas soportaban en la superficie (1/6 de la gravedad de la Tierra), pero a medida que se consume combustible y se pierde masa, la aceleración aumenta, aunque nunca llega a 1 G. Al ser tan liviana esta sección de la nave, un pequeño impulso de los propulsores de giro produce una aceleración considerable, por eso, durante el ascenso, sólo se orienta la nave utilizando propulsores que apuntan hacia abajo, nunca usa ninguno de los que apuntan hacia arriba y empujan en contra del movimiento de la nave porque se desperdiciaría combustible frenando y acelerando a la vez.

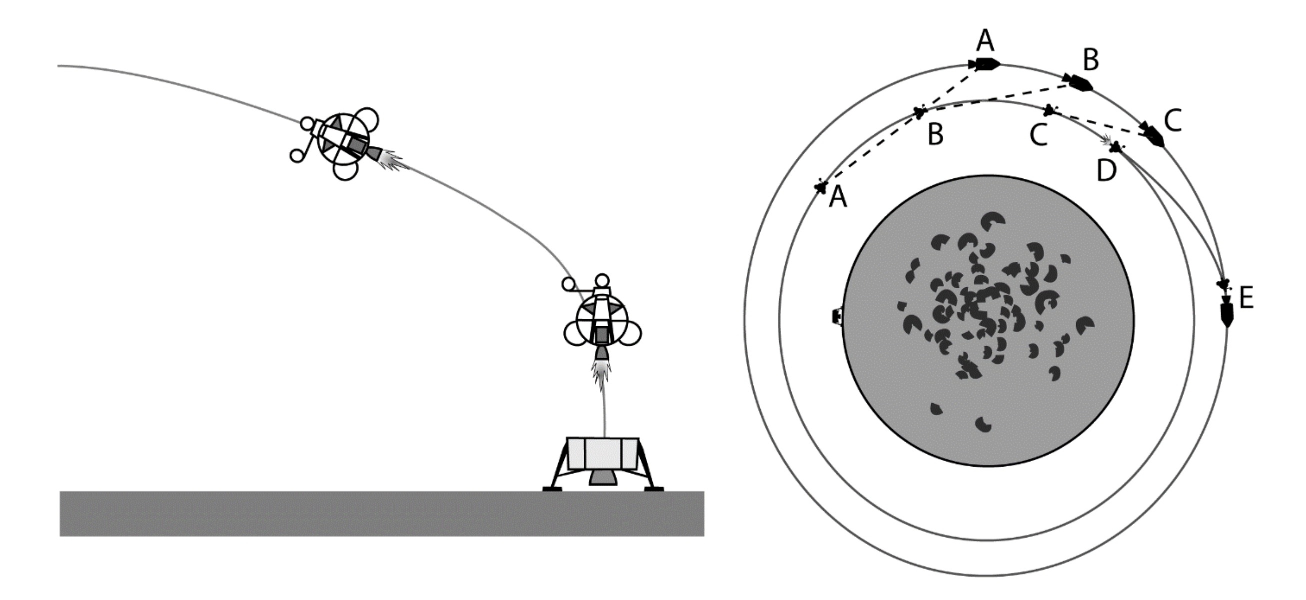


Figura 4. A la izquierda la etapa de ascenso del Módulo Lunar entrando en órbita desde la superficie lunar. A la derecha el Módulo de Comando está orbitando en una órbita más alta que la inicial del Módulo Lunar. En los momentos A, B y C el Módulo Lunar toma mediciones de la posición del Módulo de Comando y las extrapola para obtener su trayectoria. En D calcula y ejecuta una maniobra para que ambas naves coincidan en el mismo punto en el momento E.

Cabe destacar que la velocidad de rotación de la Luna sobre su eje es tan lenta (una revolución cada 27 días) que no le aporta nada significativo a la nave para ayudarla a alcanzar la velocidad orbital, a diferencia de lo que ocurre en la Tierra en donde conviene despegar en la misma dirección en que gira el planeta y cerca del Ecuador. Este ascenso dura unos 7 minutos, tiempo durante el cual los astronautas observan la información de altura y velocidad en el visor de la computadora, verificando que todo se desarrolle correctamente y al terminar la nave entra en una órbita elíptica de 17 km en la parte más baja (perilunio) y 83 km de altura en la más alta (apolunio). A partir de ahora ya no se utilizará más el motor de ascenso, todos los impulsos serán datos por los pequeños propulsores de orientación de la nave aprovechando que esta nave es muy liviana, sobre todo ahora que consumió la mayor parte de su combustible en el ascenso.

ENCUENTRO EN ÓRBITA DEL MÓDULO LUNAR Y MÓDULO DE COMANDO

Una vez en órbita, el Módulo Lunar necesita calibrar otra vez el sistema de navegación por las posibles desviaciones sufridas durante el ascenso, por eso se reorienta la nave, observando dos estrellas esta vez, y se obtiene las últimas coordenadas de la nave leídas desde Houston. Ahora que ambas naves orbitan a la Luna, pero a alturas diferentes, es necesario maniobrar hasta que se unan en órbita [11]. El Módulo de Comando durante todo este tiempo permanecerá en su órbita a 110 km de altura alrededor de la Luna sin modificar en nada su trayectoria. En cambio, el Módulo Lunar es quien deberá maniobrar para interceptar a la otra nave que se encuentra en una órbita más alta. Para lograr esto deberá primero conocer bien las coordenadas y trayectoria del Módulo de Comando y calcular una maniobra que lo ponga en una trayectoria que culmine en la intersección de ambas, por eso se calcula la trayectoria del Módulo de Comando en base a su movimiento orbital: si esta nave en algún momento disparara sus motores habría que tenerlo en cuenta para calcular la nueva trayectoria de intersección y eso requeriría mucha coordinación.

La forma más directa de encontrarse hubiera sido partir desde la superficie hacia una órbita igual a la de la nave que ya está orbitando y hacerlo en un momento exacto para que al llegar queden ambas a corta distancia. El problema es que si se suman todos los márgenes de error que existen en una maniobra así, comenzando por las coordenadas de ambas naves, sumado a que el ascenso no tiene un nivel de precisión muy alto porque se realiza con el motor encendido durante varios minutos, las desviaciones son muchas. Además, esta maniobra de encuentro entre las dos naves no necesariamente se debe realizar sólo luego de explorar la Luna partiendo en el momento adecuado, también debería poder ejecutarse en una emergencia durante el descenso en un momento cualquiera de la trayectoria.

Para obtener la información de la trayectoria del Módulo de Comando se recurre a una medición directa utilizando el radar de encuentro, que funciona emitiendo una señal que la otra nave responde automáticamente. Al obtener la señal de respuesta este radar puede calcular la distancia de hasta 750 km con una precisión de 200 m, y también obtiene la velocidad en que se mueve con respecto al Módulo Lunar. Como este radar está montado en una plataforma que gira en dos ejes que buscan automáticamente la dirección en que señal de respuesta es óptima, leyendo estos ángulos se conoce la dirección en que se encuentra el Módulo de Comando. Toda esta información se muestra en unos indicadores para que los vean los astronautas y también está conectada con la computadora para que su software la use de forma directa.

El programa para seguir al Módulo de Comando toma mediciones del radar cada un minuto, no más seguido, y con cinco mediciones, y teniendo en cuenta la posición en que estaba el Módulo Lunar al momento de tomar cada una, le alcanza para extrapolar y obtener los parámetros de una curva que se aproxima a la trayectoria que sigue el Módulo de Comando y conociendo esta curva puede predecir con cierta aproximación el estado de la nave en lo que resta de la órbita. Estas proyecciones de la trayectoria del Módulo de Comando también se podrían realizar calculando la fuerza de gravedad que afecta a la otra

nave en cada punto alrededor de la Luna y luego correr una simulación para ver en qué posición quedaría luego de, por ejemplo, una hora orbitando la Luna. El problema en este caso sería que la computadora no tiene suficiente velocidad para realizar una operación como esta en el tiempo necesario. Este tipo de simulación lo realizaban las grandes computadoras de cálculo de trayectoria en la NASA, las computadoras de las naves sólo aplicaban la simulación de la gravedad para actualizar la aceleración en cada momento, nunca realizaban proyecciones ni estimaban trayectorias de esta manera.

En las primeras misiones, antes de ir al encuentro del Módulo de Comando se dedicaba una órbita alrededor de la Luna para sincronizarse con la órbita de la otra nave pero a una altura inferior. Con la información obtenida del radar se pasaba a una órbita que era igual en forma a la del Módulo de Comando pero con una distancia constante de 28 km por debajo. Esta nueva órbita se lograba con dos maniobras, cada una con un nombre (CSI y CDH) y un programa de computadora especial. En la NASA habían practicado en el programa Gemini con encuentros en órbita siempre a 28 km de distancia constante antes de iniciar la aproximación final, por eso no se quiso innovar estando en la Luna y se prefirió dedicar unas dos horas extra a esta maniobra intermedia. En las últimas misiones, aprovechando la experiencia ganada, se eliminó este paso innecesario.

Hasta ahora en todos los demás casos nunca la computadora calculó una maniobra orbital para modificar la trayectoria de la nave, esta tarea fue realizada siempre por las computadoras de la NASA y los resultados obtenidos (hora de inicio de maniobra, dirección hacia donde apuntar y tiempo de encendido del motor) se le pasaban a la tripulación para que los cargaran en la computadora de la nave que era la encargada de ejecutar la maniobra. Esta es la única etapa del viaje en que la computadora de la nave calcula una maniobra luego de resolver cuál es la trayectoria que debe seguir para encontrarse con el Módulo de Comando. La trayectoria de intersección la obtiene con métodos similares o los usados para obtener la órbita de la otra nave, de forma puramente geométrica, sin realizar una simulación física. Sabiendo la curva de la órbita del Módulo de Comando y en qué punto de esta se encuentra en cada momento, calcula una nueva trayectoria que intersecta a la órbita de la otra nave en un punto y lo hace de forma tal de que la intersección esa se produzca en el momento en que ambas naves estén ahí. Para poder realizar este cálculo debe tener en cuenta la velocidad actual del Módulo Lunar y la nueva velocidad para entrar en la nueva maniobra. Luego debe obtener los parámetros de la maniobra para poder ejecutarla en el momento adecuado, cuando ambas naves se encuentren en un ángulo en particular en sus órbitas.

Esta trayectoria de intersección (llamada TPI por Terminal Phase I) dura unos 45 minutos y representa aproximadamente un tercio de una órbita completa alrededor de la Luna y se tienen programados dos ajustes de trayectoria cada 15 minutos, para que la computadora corrija los errores de una trayectoria tan larga y se aproxime lo más posible a lo calculado. El programa que calcula la trayectoria del Módulo de Comando por medio del radar, permanece corriendo en segundo plano constantemente para tener siempre una última estimación actualizada de la trayectoria para usar en los ajustes. Durante toda esta

trayectoria el Módulo Lunar está orbitando por debajo del Módulo de Comando, avanzando más rápido, pero, como está subiendo de órbita, cada vez avanza más lento y más cerca de la otra nave. Para compensar por posibles errores y evitar quedarse corto, cuando se calcula la trayectoria se apunta 1 Km más alto que lo necesario, así hará falta frenar al acercarse al Módulo de Comando.

Era posible realizar toda la maniobra de encuentro entre las naves aun si la computadora no funcionara o tuviera su sistema de navegación descalibrado, utilizando la computadora de backup del Módulo Lunar que también estaba conectada con el radar y realizaba sus propios cálculos de la trayectoria para realizar la intercepción. Durante toda la maniobra el piloto actualizaba la información en ambas computadoras. En caso de que este otro sistema también fallara existía una tercera opción que era navegar de forma manual y realizar las cuentas utilizando tablas pre calculadas. La información del radar se podía leer en los instrumentos y se podía obtener geométricamente en papel una trayectoria de intersección que resolviera lo leído por el radar y luego se podía iniciar la maniobra manualmente. También existía la posibilidad de que una parte o toda la maniobra se realizara desde el Módulo de Comando en lugar de desde el Módulo Lunar. El Módulo de Comando reduciría su órbita para encontrarse con el Módulo Lunar. Para estar preparado para ese caso, el piloto del Módulo de Comando durante todo el tiempo estaba activo buscando al Módulo Lunar y actualizando la información en su computadora, que también incluía una versión de los programas para realizar lo mismo que hacía la del Módulo Lunar pero en sentido inverso. Toda la información y cálculos realizados desde el Módulo Lunar, tanto los que realizaba la computadora principal como la computadora de backup, junto con la información y los cálculos que manejaba el Módulo de Comando se comparaban con los cálculos realizados en Houston para validar que se estuviera trabajando con información correcta. Todo el proceso para realizar el encuentro entre las naves era muy intenso y complejo tanto para la tripulación de ambas naves como para el personal en Tierra.

Cuando las naves estaban a menos de 3 km y se podían distinguir visualmente, el comandante tomaba el control del Módulo Lunar y realizaba manualmente varias maniobras de frenado para dirigir la nave hacia el Módulo de Comando. Como había contacto visual y el Módulo Lunar estaba muy liviano y era muy maniobrable, el comandante podía dirigir la nave manualmente pero teniendo en cuenta las características del vuelo orbital. Luego de maniobrar, el Módulo de Comando, con un peso de 16 toneladas (todavía cuenta con una parte importante de su combustible), y el Módulo Lunar, que pesa solo 2 toneladas, se encuentran a apenas metros de distancia orbitando a la misma velocidad. Desde ambas naves aprovechan para inspeccionarse visualmente entre sí, porque el resto del tiempo no es posible saber el estado en que está su parte externa. Es el piloto del Módulo de Comando quien realiza la parte final de la aproximación porque en esta maniobra es una ventaja que su nave sea la más pesada y que cada empuje produzca una aceleración más tenue y puede dar movimientos pequeños y precisos. El piloto del Módulo de Comando dirige la nave mirando a través de la ventana que está junto a la escotilla para encastrarlo a través de la

unión. El Módulo Lunar debe desactivar su piloto automático antes para que no dispare sus propulsores y se oponga al empuje del Módulo de Comando que ahora está unido a él.

Toda esta última parte se debía realizar en una parte en especial de la órbita del Módulo de Comando alrededor de la Luna, por razones relacionadas a la iluminación. Era muy importante que en ningún momento en las observaciones que debían hacerse desde alguna de las naves miraran directo hacia el Sol, porque les impediría la visión y en el caso del sextante del Módulo Lunar, enceguecería a su piloto. También era útil en caso de existir algún problema en el Módulo Lunar que desde el Módulo de Comando fuera visible una baliza luminosa que tenía encendida en esta etapa, y esto era más fácil en la parte no iluminada de la Luna. Toda la maniobra se coordinaba para que esta parte se pudiera realizar en una parte de la órbita con estas condiciones de iluminación.

Abandono del Módulo Lunar y su computadora

Estando las dos naves unidas otra vez, se hace el traspaso del material del Módulo Lunar que es necesario llevar a Tierra, sobre todo las muestras de suelo lunar y los rollos de película utilizados. Estos elementos que se transportarán tienen un peso calculado de antemano y se ubican en el Módulo de Comando en posiciones predeterminadas para que el centro de gravedad esté en un punto ya definido. Se realiza también la operación inversa y se mueve todo lo que ya no cumple un rol en el Módulo de Comando hacia el Módulo Lunar. Luego se cierra la escotilla y con una carga explosiva se detona el túnel que une a las dos naves, y se ejecuta una maniobra para alejar al Módulo de Comando. El Módulo Lunar permanece en órbita alrededor de la Luna y en condiciones ideales lo haría por siempre, pero al no realizar correcciones periódicas, irá alterando su trayectoria hasta que, transcurrido un tiempo, se estrelle contra la superficie lunar, destruyéndose junto con una de las dos computadoras que viajaban en la misión.

INYECCIÓN TRANSTERRESTRE

La para abandonar la órbita alrededor de la Luna es necesario aumentar la velocidad aproximadamente 1 km/s y para eso se debe usar el motor principal del Módulo de Servicio. Si al llegar a la Luna no hubiera funcionado este motor, la nave no habría entrado en órbita y hubiera regresado directo hacia la Tierra, pero en este caso si el motor no cumpliera su objetivo la tripulación quedaría condenada a orbitar la Luna hasta acabar con sus provisiones. Igual que en la maniobra para entrar en órbita, el motor se debía encender cuando el Modulo de Comando estaba en una posición en que no era visible desde la Tierra y eso significaba que cuando se realizara esta maniobra no habría contacto de radio, todo debía hacerse de forma autónoma.

Los parámetros para esta maniobra se transmitieron desde Houston y en su cálculo se tuvo en cuenta, además de tomar todos los últimos datos y estimaciones de la posición, velocidad y peso de la nave, el punto de llegada en la Tierra. Como durante el viaje hacia la

Tierra, esta sigue girando sobre su eje, para seleccionar una longitud específica en el planeta se debe realizar el trayecto en un tiempo tal que cuando se llegue se lo haga en ese punto. La duración de la trayectoria hacia la Tierra se selecciona con esta maniobra, mientras más largo sea el impulso dado más rápido se llegará.

Se necesita menos energía para partir de la Luna a la Tierra que de la Tierra a la Luna porque, además de estar más livianos por haber consumido unos dos tercios del combustible y haberse deshecho de una nave, el campo gravitatorio de la Luna es muy débil. Para realizar la maniobra el motor se enciende durante 2 minutos y medio y, al viajar con tan poca masa, esta vez el empuje del motor es el mismo pero produce una aceleración mayor, de aproximadamente 1 G.

Entrada en área de influencia gravitatoria terrestre

La nave se aleja de la Luna cada vez más lento porque la frena la atracción de la Luna en dirección opuesta, hasta que llega al punto de la trayectoria donde la gravedad de la Tierra es más intensa que la lunar. En este momento la nave viaja a apenas a 850 m/s (3.000 km/h), una velocidad baja para un viaje interplanetario, pero a partir de aquí comienza a ganar velocidad continuamente hacia la Tierra.

A diferencia del viaje de ida a la Luna, donde la gravedad del cuerpo más grande (la Tierra) empujaba en el sentido contrario al movimiento y el que empujaba en el sentido del viaje era el más pequeño (la Luna) y la mayor parte del trayecto se lo hacía perdiendo velocidad: en este caso se invierten las condiciones y la mayor parte del tiempo se está ganando velocidad. Como la intensidad de la atracción gravitatoria depende del inverso del cuadrado a la distancia, mientras más cerca de la Tierra se encuentra mayor es la aceleración, por eso faltando 2 horas para tocar la atmósfera la nave alcanzó una velocidad de 5,5 km/s, pero pasadas esas 2 horas, al llegar a la atmósfera, ya viaja a 11 km/s (aproximadamente 40.000 km/h), la máxima velocidad alcanzada por un ser humano.

Este trayecto es bastante similar al que se siguió para llegar a la Luna, y también es necesario realizar ajustes de dirección periódicos para mantenerse lo más cerca posible de la trayectoria ideal.

Separación del Módulo de Servicio

Cuando ya se está en la trayectoria correcta para entrar a la atmósfera terrestre, el Módulo de Servicio deja de ser necesario. No hay que volver a utilizar su gran motor y hace falta desprenderse de esta parte de la nave para entrar lo más liviano posible en la atmósfera. Deshacerse del Módulo de Servicio no es una tarea simple y no es deseable que lo sea porque sería una catástrofe que se desprendieran de esta sección de la nave en otra etapa de la misión. Durante todo el viaje la tripulación respiró el oxígeno almacenado en los tanques del Módulo de Servicio, utilizó la electricidad generada en él y mantuvo la temperatura de los sistemas eléctricos con su sistema de refrigeración por agua. El Módulo

de Comando por sí solo apenas tiene una pequeña reserva de oxígeno y en sus baterías almacena carga eléctrica para un par de horas, solamente puede ser autónomo durante un período muy breve. Los motores del Módulo de Comando están limitados a sólo cambiar su orientación por su distribución y porque que no cuentan con suficiente combustible, por esta razón no podrá ajustar su dirección una vez que se desprendan del Módulo de Servicio.

Para iniciar la secuencia de separación, la tripulación primero debe cerrar manualmente las válvulas de las cañerías de agua y de oxígeno que la unen al Módulo de Servicio y mover interruptores en el tablero eléctrico para tomar de ahora en adelante la alimentación eléctrica de las baterías del Módulo de Comando. Luego se enciende un interruptor en el panel del Módulo de Comando que activa el sistema automático de separación ubicado en el Módulo de Servicio, que, una vez iniciado, se encarga de detonar unas cargas explosivas que cortan todas las uniones entre los dos módulos, y luego realiza una maniobra que aleja al Módulo de Servicio de la nave con los astronautas, maniobra que solamente puede realizar este porque es el único con capacidad para controlar su desplazamiento en el espacio. Este sistema automático no es una computadora sino un circuito eléctrico que cumple esta sola función.

Ahora el Módulo de Comando vuela solo en dirección a la atmósfera terrestre. Al desprenderse del Módulo de Servicio quedó expuesto el escudo térmico que se encuentra en la parte donde estaban unidos los dos módulos. La nave se reorienta para que el escudo quede apuntando hacia delante, para poder enfrentar la atmósfera con él.

Reentrada en la atmósfera

El Módulo de Comando, con un peso de aproximadamente 6 toneladas, se acerca a la Tierra, no directamente hacia ella sino de forma tangencial, hacia su horizonte, y llega a alcanzar una velocidad de alrededor de 11 km/s. Estando cerca necesita dos cosas: reducir la velocidad para depositarse suavemente en el mar y además debe controlar el vuelo para culminar en un punto específico de la Tierra donde los están esperando. Lo que queda de la nave no cuenta con ningún motor para frenar ni para desplazarse a voluntad, deberá valerse de la atmósfera para que le reduzca la velocidad y para que le de sustentación y le permita planear hacia su destino.

La reentrada comienza cuando la nave, que viene desde la Luna acelerando constantemente sin que nada la frene en el vacío del espacio, entra en la atmósfera terrestre que opone una gran resistencia a su movimiento. No existe una altitud en la Tierra donde se divida entre atmósfera y vacío: es un límite difuso. La densidad de la atmósfera decae constantemente y a alturas de 300 km ya es casi inexistente aunque aún hay partículas de gas dispersas. La resistencia que opone la atmósfera a la nave depende de la densidad del aire (es directamente proporcional, a mayor densidad mayor resistencia) y de la velocidad de la nave (aumenta con el cuadrado de la velocidad con que la nave se mueve con respecto al aire, si la velocidad aumenta al doble, la fuerza de resistencia se multiplica por cuatro). Cuando la nave se cruza con la atmósfera lo hace al principio en lugares de muy baja

densidad, por eso, a pesar de ir con una velocidad muy alta, la resistencia no es de magnitud apreciable. A medida que pasa a altitudes con densidades de aire mayores, la gran velocidad de la nave hace que la fuerza que se le opone a su movimiento sea muy elevada, llegando en su punto máximo a una desaceleración de 7 G, mayor a cualquier aceleración creada por el cohete Saturno V u otra que haya sufrido la tripulación durante toda la misión. Luego de pasar ese máximo de frenado sigue aumentando la densidad del aire a medida que llega a capas más bajas de la atmósfera, pero como ya perdió mucha velocidad lo hace con una resistencia mucho menor.

La nave recibe esta fuerza de frenado del aire sobre el escudo térmico, la parte más ancha de la nave, y los astronautas están sentados en el interior de la nave mirando en sentido contrario, por eso la fuerza de frenado se les aplica sobre sus espaldas, igual que cuando despegan: es la posición en la que se reciben todas las fuerzas importantes dentro de la nave. Todo el reingreso se tiene que hacer por instrumentos por varios motivos relacionados a la complejidad y precisión necesaria, y a esto se suma que los tripulantes no pueden ver por la ventana hacia donde se dirigen sino sólo hacia atrás, y no tienen forma de controlar la maniobra visualmente. Fuera de la nave, a causa de la enorme resistencia del aire, se alcanzan temperaturas muy elevadas cercanas a los 3.000 grados. El escudo térmico, de gran resistencia estructural, protege a la nave de la fuerza del aire y también, como su nombre lo indica, del gran calor producido, y, de hecho, dentro de la nave no se nota ningún aumento de la temperatura.

El Módulo de Comando puede controlar la dirección en que se mueve en la atmósfera rotando por medio de sus motores de giro. La nave, igual que un avión en sus alas, recibe una fuerza de sustentación del aire que se mueve a través de ella porque no se enfrenta al aire completamente de frente. Externamente la nave es simétrica pero su centro de gravedad no está en el eje central sino desplazado hacia donde se encuentran los astronautas. El aire que choca contra el escudo hace que este, y toda la nave, avance en un ángulo que produce que por una mitad de la nave circule más aire que por la otra y eso causa una diferencia de fuerzas que le da sustentación. Rotando la nave sobre su eje se elige de qué lado se produce la sustentación, así se puede lograr que la nave se eleve, descienda o se mueva hacia los lados.

Como se vio anteriormente, la comunicación por radio es más complicada cuando la nave está cerca del planeta porque se necesitan muchas repetidoras a lo largo de la trayectoria sobre la superficie. A esto se suma que el gran calor que rodea a la nave ioniza el aire y produce un escudo de plasma que interfiere a las ondas de radio e impide transmitir o recibir señales. Por eso al poco de iniciar la reentrada y durante más de tres minutos se produce un estado de blackout (apagón) que hace que durante gran parte de la maniobra de reentrada no haya comunicación entre Houston y la nave, y por eso toda la maniobra debería poder realizarse sin ayuda externa.

Cuando recién se entra en la atmósfera lo primero que se debe realizar es bajar la velocidad por debajo de la velocidad orbital (8 km/s) porque viajando por encima de esta velocidad existe el riesgo, en caso de que, si por algún problema la nave volviera salir de la

atmósfera, o bien quedaría orbitando o saldría en una trayectoria muy larga que duraría horas o hasta años y no habría posibilidad de salvar a la tripulación en una situación como esa. Una vez que la velocidad se redujo por debajo de este límite existe la certeza de que pase lo que pase la nave volverá a la superficie terrestre en poco tiempo. Durante los primeros dos minutos se logra esto siguiendo una trayectoria que produce en su momento más intenso 7 G pero que la mayor parte del tiempo no baja de 4 G.

Luego de reducir la velocidad inicial, la nave vuelve a ascender hasta zonas de muy baja densidad y el tiempo en que permanezca en esa trayectoria dependerá del punto de amerizaje seleccionado. Mientras más lejos esté el destino para amerizar más tiempo se deberá pasar fuera de la atmósfera para hacer una trayectoria más larga. Esto le da mayor flexibilidad a la flota que los espera cerca del punto de llegada para evitar zonas con malas condiciones hasta horas antes de que llegue la nave.

Secuencia de reentrada

A diferencia del viaje en el vacío del espacio, donde la física de los campos gravitatorios es muy predecible, al desplazarse en la atmósfera existen muchos factores indeterminados con anterioridad. La densidad del aire no es pareja y hay muchas variaciones dependientes de factores climáticos. Las maniobras no se pueden realizar haciendo ajustes cada varios minutos, como en el espacio, sino que constantemente se tiene que monitorear las fuerzas que el aire ejerce sobre la nave. Esta es una tarea de procesamiento intensivo para la computadora que debe ajustar lo más rápidamente posible las desviaciones de la trayectoria.

La trayectoria con que viene la nave antes de tocar la atmósfera es vital para realizar la reentrada. El ángulo con el que toca la atmósfera debe ser el correcto (6,5 grados) para poder moverse en ese medio de forma apropiada. Si hubiera una desviación hacia fuera de la Tierra, la nave podría cruzar una región de la atmósfera sin perder la suficiente velocidad y luego volver a salir al vacío, sin posibilidad de regresar a causa de su gran velocidad (Figura 5.a). Si en cambio, la trayectoria estuviera desviada más hacia la Tierra, la resistencia del aire puede aumentar muy rápido porque se llegaría a zonas de alta densidad antes de perder la suficiente velocidad. La fuerza se haría mayor que lo que puede resistir la nave o se dirigiría muy rápido hacia la superficie sin posibilidad de controlar el vuelo (Figura 5.b). Por eso el sistema de navegación debe garantizar una trayectoria correcta con precisión de fracción de grado para continuar la reentrada (Figura 5.c).

Para dar comienzo a la secuencia de reentrada se toma como entrada en la atmósfera el momento en que los acelerómetros miden una tenue aceleración de 0,05 G producida por la resistencia del aire y esto debería ocurrir a una altura de aproximadamente 120 km. Al cruzar este límite, que es un punto arbitrario, la computadora comienza a operar en modo reentrada en atmósfera (Figura 6.a). Durante varios segundos la nave sigue su trayectoria sin controlar su dirección mientras aumenta la fuerza de frenado. Cuando llega a

un punto donde la resistencia se hace de 2 G, el empuje del aire es suficiente para permitir controlar el vuelo (Figura 6.b).

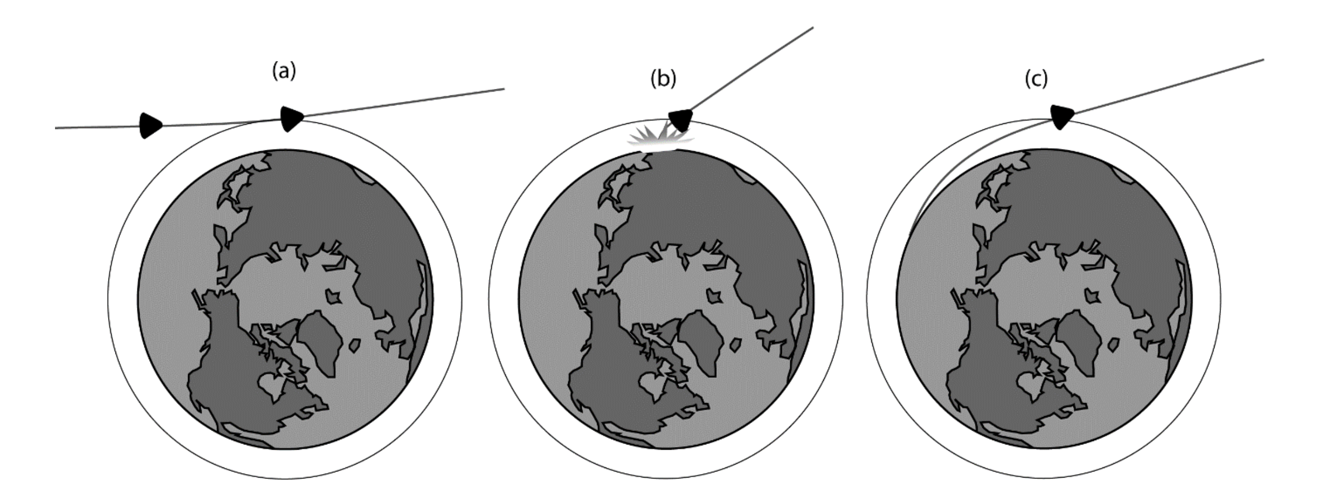


Figura 5. Distintos ángulos de entrada a la atmósfera. (a) Si el ángulo está muy desviado hacia afuera la nave puede revotar y entrar en órbita. (b) El ángulo está muy desviado hacia la Tierra y la nave se destruye por las fuerzas excesivas. (c) Se entra con el ángulo de 6,5 grados y se realiza la reentrada.

La computadora dirige a la nave rotándola sobre su eje para variar el ángulo con el que el escudo se enfrenta al aire. Si solamente se tiene en cuenta la dirección vertical de la nave entonces hay dos ángulos posibles para orientar la nave, uno con el centro de gravedad lo más bajo posible y el otro con este en lo más alto, cada uno se logra rotando la nave 180 grados con respecto al otro. Para poder seleccionar ángulos intermedios, que no asciendan tanto o no desciendan tanto, se debe rotar la nave para que el centro de gravedad este más inclinado hacia un lado, pero en este caso la nave va a desviar su trayectoria en el eje horizontal, no solo en el vertical. Para evitar estas desviaciones y no alejarse mucho del punto de destino, la computadora debe compensar y si, por ejemplo, estuvo 10 segundos con una rotación de 60 grados hacia la izquierda, en los siguientes 10 segundos la rota 120 grados para que quede 60 grados hacia la derecha y que de esta forma se anule la desviación. Si se quisiera variar la dirección en el eje horizontal alcanzaría con no equilibrar estas rotaciones.

Lo primero que hace la computadora al tomar el control de la maniobra es llevarlo en una trayectoria que haga pasar a la nave por capas cada vez más densas hasta alcanzar el máximo de 7 G (Figura 6.c) y luego la hace ascender para que siga perdiendo velocidad pero no de forma tan brusca. Cuando este ascenso culmine, la nave ya debe haberse frenado lo suficiente y estar por debajo de la velocidad orbital (Figura 6.d), para luego salir a zonas poco densas de la atmósfera donde ya no se puede controlar el vuelo (Figura 6.e). Durante esta primera etapa de vuelo controlado la computadora dirigió a la nave para que saliera de la atmósfera con determinada velocidad y dirección y así acotó la zona donde terminará la trayectoria. Cuando recién tocó la atmósfera existía una zona de miles de kilómetros donde

podía terminar la nave siguiendo trayectorias válidas, pero ahora sólo puede terminar en una zona de cientos de kilómetros y lo que sigue es el ajuste fino.

La gravedad y la tenue resistencia del aire vuelven a atraer a la nave que realiza su segunda reentrada, pero a mucha menor velocidad (Figura 6.f). Cuando la trayectoria vuelve a pasar por un punto de 2 G de resistencia se pasa la segunda parte de vuelo controlado (Figura 6.g). La computadora ahora controla el vuelo para dirigir la nave lo más cerca posible del destino.

El EMS (Entry Monitor System) estaba pensado para utilizase durante la reentrada, y, aunque se le daba otros usos durante la misión, es ahora el momento para aprovecharlo. Ni bien se inicia la maniobra de reentrada, se configura el visor para mostrar la distancia recorrida (en millas náuticas) y se le carga la distancia hacia el objetivo en el mar para que muestre todo el tiempo cuanto falta recorrer para llegar allí. Otro visor del EMS, que sólo se utilizaba en esta etapa, era una representación gráfica de unos 9 cm de ancho del progreso de la reentrada. Una cinta de mylar se iba desplazando horizontalmente mientras un punzón le dejaba una marca que iba dibujando una línea a medida que se movía el rollo. El eje horizontal, el movimiento de la cinta, representaba la velocidad de la nave con respecto a la Tierra, a medida que se reducía la velocidad la cinta avanzaba, mostrando cuanta velocidad faltaba para detenerse. En el eje vertical, el marcador que escribía, representaba la tasa de desaceleración de la nave. Analizando este gráfico a medida que se dibujaba, fácilmente se podía detectar si algo estaba saliendo mal, si se estaba desacelerando de más o de menos. En caso de decidir que se estaba siguiendo un rumbo erróneo, el comandante podía tomar el control manual del Módulo de Comando y hacerlo rotar para elevarse o descender más y así manipular la desaceleración que le correspondía en cada momento. En la cinta del EMS venían dibujadas varias líneas que mostraban las trayectorias con la fuerza de desaceleración máxima y mínima para facilitar la detección de trayectorias que se fueran de rango, y además se mostraba las trayectorias ideales para los distintos tipos de objetivos a alcanzar, y así se podía controlar manualmente la nave manteniéndola cerca de esa línea mientras se miraba este instrumento.

AMERIZAJE

Una vez que la nave perdió la mayor parte de la velocidad inicial, llegando a menos de 1.000 km/h, ya no tiene sustentación del aire y su vuelo no es controlable. Entra en la parte balística de la trayectoria de reentrada y, si todo salió correctamente, se dirige directo hacia el destino planificado (Figura 6.h). Estando a 10 km de altura sobre el punto de amerizaje la tripulación desactiva manualmente los motores del Módulo de Comando para que no vuelvan a intentar controlar la orientación de la nave y también activan el sistema automático de amerizaje. Este sistema es independiente de la computadora e inicia una secuencia de eventos que funcionan por medio de secuenciadores electromecánicos y barómetros que miden la altitud de la nave. A una altura de 7 km se despliegan dos pequeños paracaídas que cumplen la tarea de frenar un poco la caída, pero, principalmente,

orientan la nave con el escudo hacia abajo y la escotilla hacia arriba. Se comienzan a abrir las válvulas para que se iguale la presión interna de la nave con la atmósfera, y, finalmente, a los 3 km de altura se despliegan tres enormes paracaídas que harán que la nave reduzca su velocidad de caída y, aunque alcanzaría con sólo dos, el tercero es para mayor seguridad (Figura 6.i). Al tocar el mar el Módulo de Comando cae a 30 km/h, una velocidad que destruiría a la nave y a sus tripulantes chocando en tierra firme, pero sobre el agua representa apenas un golpe (Figura 6.j).

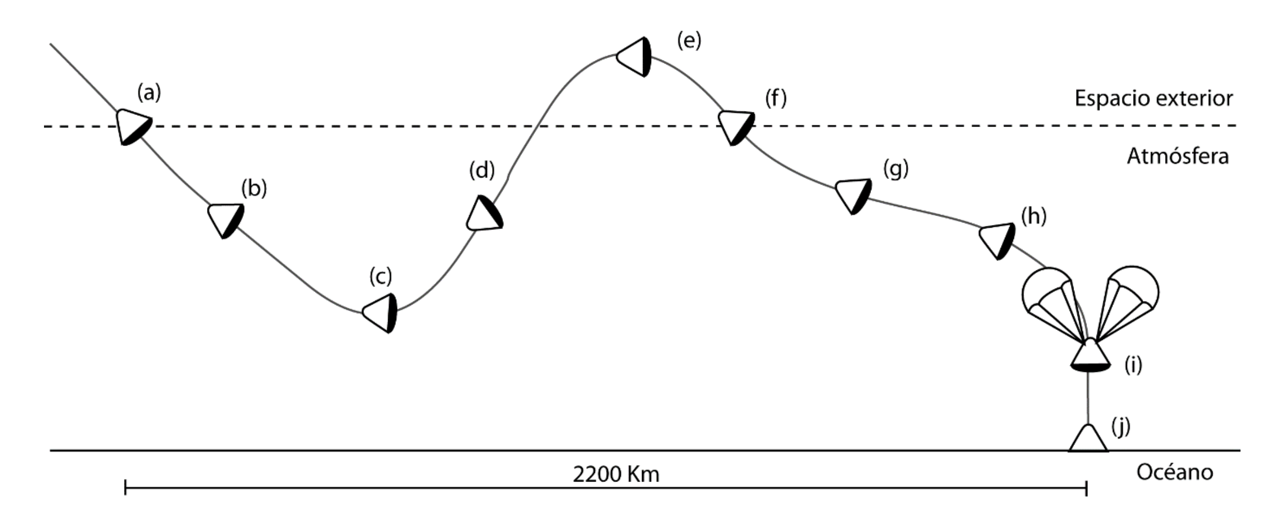


Figura 6. (a) Reentrada en la atmósfera.(b) Fuerza de sustentación y control del vuelo.(c) Máxima fuerza de frenado. (d) La velocidad se redujo por debajo de la orbital.(e) Salida al espacio.(f) Segunda reentrada.(g) Segunda etapa de vuelo controlado.(h) Trayectoria balística.(i) Apertura de paracaídas. (j) Amerizaje.

El Módulo de Comando y sus tres tripulantes, lo único que quedó del enorme cohete Saturno V que partió desde Cabo Cañaveral, permanecen flotando en el mar. Fue tan preciso el amerizaje que están, a lo sumo, a unos pocos kilómetros del portaviones que los espera y que envía helicópteros a auxiliarlos y así concluir con la misión.

Salvo los últimos kilómetros que se realizaron en paracaídas y el ascenso, controlado por la computadora del Saturno V, la computadora del Módulo de Comando y del Módulo Lunar controlaron la mayor parte de las muy variadas maniobras de la misión.

CONCLUSIONES

Hicimos un recorrido completo de una misión típica hacia la Luna, y, de entre los innumerables detalles, nos centramos en los aspectos relacionados con el control de la misión por medio de la computadora. Misiones como esta se realizaron desde 1967 a 1972 y en la siguiente parte se analizan de qué manera se desarrollaron en la historia.

BIBLIOGRAFÍA

[1] W. D. Woods (2011). *How Apollo Flew to the Moon*, Springer Praxis.
[2] IBM (2015). *The Apollo Missions.* http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/apollo/ (10 de junio de 2016).

[3] W. Haeussermann (1970). *Description and performance of the Saturn Launch Vehicle's navigation, guidance and control system*, NASA.
[4] IBM (2015). *Saturn Instruments Unit.* http://www-03.ibm.com/ibm/history/exhibits/space/space_saturn2.html (9 de junio de 2016)
[5] M. S. Johnson (1971). *MIT's role in project Apollo, Volume V: The software effort*, Charles Stark Draper Laboratory, Marzo 1971.
[6] Jet Propulsion Laboratory (2016). *Deep Space Network.* http://deepspace.jpl.nasa.gov/ (6 de junio de 2016).
[7] L. Hutchinson (2012). *Going boldly: Behind the scenes at NASA's hallowed Mission Control Center* http://arstechnica.com/science/2012/10/going-boldly-what-it-was-like-to-be-an-apollo-flight-controller/1/ (20 de mayo de 2016).
[8] NASA y AV Corporation (prod.) (1970). *This Is Mission Control*,NASA
[9] D. A. Mindell (2008). *Digital Apollo*, MIT Press.
[10] NASA (2013). The Apollo Lunar Module Alignment Optical Telescope. En: *Apollo Lunar Surface Journal*. https://www.hq.nasa.gov/alsj/aot.htm (11 de junio de 2016).
[11] F. O'Brien (2005). Lunar Orbit Rendezvous. En: *The Apollo Flight Journal*. http://history.nasa.gov/afj/loressay.htm (5 de junio de 2016).